

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO DE INGENIERÍA BIOMÉDICA
TRABAJO FIN DE GRADO**

**Resolución numérica con Matlab de Problemas
de Ingeniería Biomédica**

Amparo Güemes González

2016

TRABAJO FIN DE GRADO

TÍTULO: Resolución numérica con Matlab de Problemas de Ingeniería Biomédica

AUTOR: Amparo Güemes González

TUTOR: Francisco José Navarro Valero

DEPARTAMENTO: Departamento de Matemática Aplicada a las Tecnologías de la Información y las Comunicaciones

TRIBUNAL:

Presidente: José María Goicolea Ruigómez

Vocal: Francisco José Navarro Valero

Secretario: Javier Jesús Lapazarán Izargain

Suplente: María Luisa Cuadrado Ebrero

FECHA DE LECTURA: _____

CALIFICACIÓN: _____

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO DE INGENIERÍA BIOMÉDICA
TRABAJO FIN DE GRADO**

**Resolución numérica con Matlab de
Problemas de Ingeniería Biomédica**

**Amparo Güemes González
2016**

Resumen

Hasta este curso 2015-2016, las prácticas disponibles en la asignatura Métodos Numéricos en Biomedicina (MNUB, del Grado en Ingeniería Biomédica, GIB) consistían en la resolución de problemas generales mediante métodos numéricos, sin ninguna referencia concreta a la Ingeniería Biomédica (con excepción de la parte del temario de métodos de elementos finitos, que incluye aplicaciones en biomecánica).

El objetivo de este Trabajo de Fin de Grado es diseñar e implementar un conjunto completo de prácticas que cubran los contenidos matemáticos de las prácticas actualmente disponibles aplicándolos a la resolución de problemas específicos de la ingeniería biomédica. Estas prácticas se implementan en Matlab, del que la UPM dispone la licencia de campus.

Las prácticas van precedidas de un planteamiento de cada problema biomédico. Este planteamiento incluye la deducción del modelo matemático que representa el problema en cuestión, salvo que sea excesivamente complicado (en comparación con el nivel exigible en el GIB), en cuyo caso se realizará una introducción teórica del proceso físico-químico a estudiar. Lo que se busca es que los problemas sean representativos de los temas estudiados a lo largo del grado en otras asignaturas. Las prácticas incluyen además un código Matlab ya escrito (total o parcialmente) o simplemente las instrucciones para la escritura del código por parte del alumno.

Lo que se pretende con estas prácticas es reforzar el aprendizaje del alumno, tanto en sus aspectos de planteamiento/modelización de problemas, como en los de resolución, presentación escrita/gráfica de resultados y análisis de los mismos.

Para lograr los objetivos expuestos se ha realizado en primer lugar una exhaustiva revisión bibliográfica sobre el tema, seguido del diseño de las prácticas, su implementación en Matlab y la prueba de los códigos. Una vez verificado su correcto funcionamiento, se redactó una guía del alumno, que contiene tanto el planteamiento teórico de la práctica como las instrucciones para

su realización, y una guía del profesor, que incluye las soluciones de las prácticas y, en su caso, los problemas más habituales esperados en la resolución de las mismas. Se pretende con esta guía del profesor disponer de un manual que pueda ser fácilmente utilizado por posibles monitores de prácticas que ayuden al docente en su labor durante las sesiones de laboratorio de la asignatura.

Summary

Until the current academic year (2015-2016), the available laboratory exercises for the course Numerical Methods in Biomedicine (MNUB, Grade in Biomedical Engineering, GIB) consisted on solving general problems applying numerical methods, without any concrete reference to Biomedical Engineering (except for the lessons based on finite elements methods, which includes applications in biomechanics).

This Bachelor Final Project goal is to design and implement a complete set of laboratory exercises covering the same mathematical contents of the current lab exercises but applied to solve specific biomedical engineering problems. These lab exercises are implemented in Matlab, whose campus license is available to the UPM.

The aim was to find representative problems of the topics studied in other courses of the degree. Every practice is preceded by a description of the relevant biomedical problem. The exposition includes the mathematical model deduction of the problem, unless it is excessively complicated (compared to the expected level in GIB). In such cases, a theoretical introduction of the chemical or physical process is provided. The practices also include a code, which is already written in Matlab (total or partially), or just the instructions for the student to write it.

The underlying purpose of these lab exercises is to support the students' learning process in several important aspects: problem formulation and modelling, its solution and the analysis of results given as data tables or graphics.

In order to seek the most illustrative problems, an exhaustive bibliographical review on each topic was carried out. It was followed by the design of the lab

exercises, their implementation in Matlab and the code testing. As soon as their correct functioning was verified, a student guide was written, which contains both the theoretical description of the problem and the instructions for its solution. A teacher's guide is also provided, which includes the problems' answers and the most common difficulties expected in the resolution, where required. This guide attempts to be a teacher's manual that can be easily used by possible laboratory assistants to help the teacher in his work during the practical sessions of the course.

Palabras clave

Métodos numéricos, Ingeniería Biomédica, Matlab, guía, problemas y aplicaciones biomédicas.

Keywords

Numerical methods, Biomedical Engineering, Matlab, guide, Biomedical problems and applications.

Índice

Introducción	1
Motivación y objetivos	1
Formato	2
Práctica 1	4
Cuaderno de enunciados	5
Cuaderno de soluciones	9
Cuaderno de entrega	14
Práctica 2	15
Cuaderno de enunciados	16
Cuaderno de soluciones	19
Cuaderno de entrega	23
Práctica 3	24
Cuaderno de enunciados	25
Cuaderno de soluciones	28
Cuaderno de entrega	32
Práctica 4	33
Cuaderno de enunciados	34
Cuaderno de soluciones	39
Cuaderno de entrega	42
Práctica 5	43
Cuaderno de enunciados	44
Cuaderno de soluciones	48
Cuaderno de entrega	53
Práctica 6	54
Cuaderno de enunciados	55
Cuaderno de soluciones	58
Cuaderno de entrega	60
Práctica 7	61
Cuaderno de enunciados	62
Cuaderno de soluciones	66
Cuaderno de entrega	70
Práctica 8	71
Cuaderno de enunciados	72
Cuaderno de soluciones	75
Cuaderno de entrega	77
Bibliografía	78

Introducción

La Ingeniería Biomédica se define como la aplicación de los principios y técnicas de la ingeniería al campo de la medicina. Es un trabajo multidisciplinar que aplica la experiencia adquirida en diversos campos de estudio para solventar y reducir las necesidades médicas, obteniendo así beneficios en la salud de las personas.

Uno de los retos principales de esta ingeniería es el enorme rango de problemas que lleva asociados, que se extienden desde la escala molecular hasta la macroscópica. Éstos se han tratado de solventar desde distintas ramas de conocimiento: mediante circuitos eléctricos, mecánica del sólido y de fluidos, termodinámica, fenómenos de transporte químico y bioquímico,... La labor del ingeniero biomédico es unificar todas estas aproximaciones a las soluciones de los problemas.

Las matemáticas siempre han constituido una herramienta eficaz para poder modelar los problemas ingenieriles que describen situaciones del mundo real. En la actualidad, su aplicación en el campo de la biomedicina se está viendo ampliada para tratar problemas como los de estimar parámetros que no se pueden medir directamente, realizar simulaciones de procesos hormonales o analizar los efectos de medicamentos sin necesidad de testarlo en animales *in vivo*, entre otros. El objetivo es estudiar de manera cuantitativa cómo se controlan los sistemas fisiológicos y fisiopatológicos, es decir, entender la dinámica de los procesos biológicos. Para ello, por lo general se describe el comportamiento biológico algebraicamente, en forma de polinomios, integrales o ecuaciones diferenciales, como se mostrará en los problemas a lo largo del trabajo.

La dificultad se halla, sin embargo, en que la mayoría de estos problemas no pueden resolverse analíticamente o de manera exacta, de forma que su solución debe ser abordada con ayuda de algún procedimiento numérico.

La Ingeniería Biomédica posee algunas características que la convierten en un campo importante de aplicación de los métodos numéricos. En primer lugar, las mediciones de propiedades biológicas, físicas o químicas no son exactas, es decir, se cometen errores y por ello no basta con realizar una sola medida de la propiedad de interés, sino que son necesarias un conjunto de medidas con los inconvenientes que ello conlleva: variabilidad de las condiciones de medición y ruido, descalibración de los aparatos de medida, obtención de un conjunto de datos discretos con los que a priori es difícil trabajar,... En segundo lugar, lo que se busca es obtener unos modelos que sean lo más exactos posible, es decir, que minimicen los errores, así como conseguir una ejecución computacional que conduzca a una solución convergente y estable. Todos estos aspectos son claves en la elección de los algoritmos numéricos a emplear.

Motivación y objetivos

El objetivo de este Trabajo de Fin de Grado es plantear los problemas de las sesiones prácticas de la asignatura troncal Métodos Numéricos en Biomedicina, impartida en el primer cuatrimestre del tercer curso del Grado de Ingeniería Biomédica de la Universidad Politécnica de Madrid.

Hasta la fecha, estas clases prácticas consistían en resolver, empleando los métodos numéricos estudiados en la asignatura, diversos problemas de carácter abstracto o no estrictamente vinculados con el campo de la biomedicina. Estas prácticas se realizaban usando MATLAB como plataforma de programación. Aunque los alumnos aprendían los aspectos básicos de los algoritmos numéricos utilizados, no tenían clara su aplicación para resolver los problemas biológicos y fisiológicos que realmente les interesaban. Lo que se busca con este trabajo es promover la motivación de los estudiantes de ingeniería biomédica, mediante la aplicación en los ejercicios planteados de los conocimientos adquiridos en otras asignaturas del grado.

De acuerdo con este planteamiento, algunos de los ejercicios prácticos se readaptaron, a la vez que se añadieron otros completamente nuevos para proporcionar ejemplos de procesos biológicos, fisiológicos y químicos reales que se resuelven con métodos numéricos.

Como entorno de programación para la resolución de problemas se ha empleado MATLAB, ya que se trata de una herramienta de fácil manejo que está disponible para todo tipo de sistemas operativos. Se trata además de un software muy usado en universidades y centros de investigación de todo el mundo, lo cual genera un valor añadido para utilizarlo como plataforma de enseñanza, ya que los alumnos estarán preparados para emplearlo en futuros trabajos.

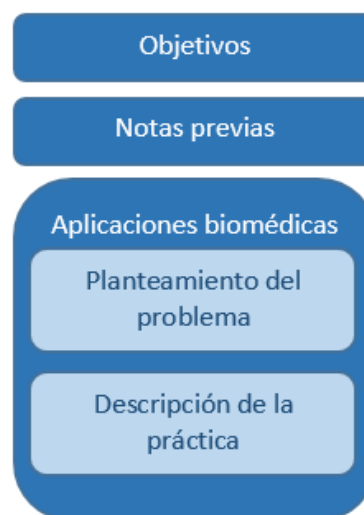
Formato

El trabajo está constituido por un total de ocho prácticas en las que se aplican los conocimientos adquiridos en las correspondientes clases teóricas de la asignatura.

Cada práctica está constituida por un cuaderno de enunciados para el alumno, un cuaderno de soluciones para el profesor y un cuaderno de entrega de los resultados obtenidos, a rellenar por el alumno durante la sesión.

En cada cuaderno de enunciados se desarrollan una o dos aplicaciones biomédicas en las que se muestra un problema relacionado con esta disciplina. El formato de cada práctica es siempre el mismo: se proporciona una primera introducción general, indicando los objetivos de la práctica y resaltando los aspectos relevantes del tema que se va a tratar. A continuación, en un apartado titulado “Notas previas”, se comentan los aspectos más relevantes a la hora de solventar los problemas y se indican las instrucciones referidas a la entrega de resultados. El esquema de la figura muestra los apartados principales de cada una de las prácticas.

Para cada aplicación biomédica se ofrece un planteamiento teórico del mismo, que sirve como introducción al sistema que se va a estudiar y constituye una verdadera motivación para los estudiantes, al mostrar algunas de las aplicaciones más comunes en el campo de la ingeniería biomédica de los métodos estudiados en el aula. En estas descripciones se hace especial hincapié en las dificultades a las que se enfrentan los ingenieros biomédicos a la hora de



realizar mediciones y de modelar el comportamiento de los procesos de interés. Cabe destacar que los problemas seleccionados son muy relevantes para el estudiante, puesto que ha lidiado con ellos en otras asignaturas del grado pero desde puntos de vista distintos al matemático.

En las prácticas en las que es necesario profundizar en algún aspecto de la teoría explicada en el aula se proporciona también una breve introducción matemática.

Además del planteamiento del problema, se proporciona una detallada descripción de los pasos a seguir en el desarrollo de la práctica. En las primeras prácticas, el alumno debe ejecutar un código proporcionado, con el algoritmo ya implementado. Por ello, estas prácticas están más orientadas al análisis de los resultados obtenidos, relacionándolos con el proceso biológico subyacente. A medida que avanzan las prácticas y los alumnos adquieren mayor destreza en el manejo de MATLAB, se pide que elaboren pequeños programas implementando los algoritmos necesarios para resolver los problemas planteados.

La siguiente tabla resume los problemas biomédicos planteados asociados a los temas explicados en las sesiones teóricas.

Número de práctica	Tema	Aplicación biomédica 1	Aplicación biomédica 2
Práctica 1	Errores	Epidemiología	Concentración de un fármaco en función del tiempo
Práctica 2	Sistemas de ecuaciones lineales	Distribución de tensiones y esfuerzos	Balances de masa metabólicos
Práctica 3	Ecuaciones no lineales	Permeabilidad de fluidos en hidrogeles	Deformación plástica en biomateriales
Práctica 4	Ajuste por mínimos cuadrados	Desintegración radiactiva	Curva disociación hemoglobina
Práctica 5	Interpolación	Fermentación de la penicilina	Temperatura corporal de un paciente
Práctica 6	Integración numérica	Administración de solución salina por vía intravenosa	
Práctica 7	Integración numérica de problemas de valor inicial	Unión monovalente del ligando a los receptores de membrana celulares	Infección viral
Práctica 8	Ecuaciones en derivadas parciales	Difusión simple de moléculas a través de una membrana plana	

En algunas prácticas se añaden preguntas etiquetadas como como “Actividades previas”, que deben ser resueltas antes de la sesión de laboratorio con el fin de afianzar conceptos y obtener un mayor rendimiento de las prácticas.

En el cuaderno de soluciones del profesor se proporcionan las soluciones numéricas y gráficas de los problemas, así como posibles explicaciones a las preguntas planteadas sobre análisis de los resultados.

Práctica 1

Errores

Aplicaciones biomédicas

- 📌 Epidemiología
- 📌 Concentración de un fármaco en función del tiempo

Práctica 1

ERRORES

Objetivos

El objetivo de esta práctica es entender y aplicar, de forma práctica, lo aprendido sobre errores en el tema 2 de la asignatura.

Se ilustrarán las limitaciones en la representación de los números en un ordenador, usando un número finito de dígitos, así como algunos efectos de la inexactitud de la aritmética finita de las máquinas.

Esto será de gran importancia en las prácticas posteriores, puesto que este tipo de errores estarán presentes en prácticamente todas las operaciones que se realicen en una máquina.

Actividades previas

Antes de la sesión es necesario que:

- Lea y comprenda la práctica. De esta manera aprovechará de manera más eficiente las horas de laboratorio.
- Realice las tareas señalados como **actividad previa**.

Nota previa

Las respuestas a las cuestiones planteadas, gráficos de soluciones, etc. deben rellenarse y entregarse en el cuaderno de entrega de soluciones disponible en el Moodle de la asignatura.

Ejercicio 1

El objetivo es analizar los errores que se producen al calcular las raíces de una ecuación cuadrática empleando las distintas formas de cálculo presentadas en el aula.

Aplicación biomédica: epidemiología

Planteamiento del problema

Un problema importante de la biología y de la medicina es el que se ocupa de la ocurrencia, propagación y control de una enfermedad contagiosa, esto es, una enfermedad que puede transmitirse de un individuo a otro. La ciencia que estudia este problema se llama *epidemiología*.

Los problemas asociados a la propagación de una enfermedad pueden ser bastante complicados, por lo que interesa disponer, como primera aproximación, de un modelo matemático sencillo que describa la propagación de una enfermedad.

Supongamos que en una población tenemos un conjunto de N personas infectadas por una cierta enfermedad. Supongamos, además, que la tasa de cambio del número de personas infectadas viene dado por una función cuadrática

$$dN/dt = f(N) = aN^2 + bN + c$$

donde a , b y c son constantes.

Si estamos interesados en conocer si, para los valores de a , b , c dados, se llegará a estabilizar en algún momento el número de personas infectadas (es decir, $dN/dt = 0$), la respuesta a esta pregunta vendrá dada por la existencia de raíces reales positivas de la ecuación cuadrática

$$aN^2 + bN + c = 0.$$

Descripción de la práctica

Sean $a = 0.0000001$, $b = 0.123456789$, $c = 0.0000001$.

- Codifique un m-file, de nombre GRxx_P1_E1.m, que utilice **format long**, y que calcule las raíces de la ecuación cuadrática en la forma ordinaria y, usando la función **sign**, en la forma recomendada que combina la ordinaria para una raíz y la racionalizada para la otra.
- Calcúlelas también, en el mismo m-file, usando la función **roots** de Matlab.
- ¿Qué método usa Matlab? ¿Alguno de los anteriores u otro distinto? (sugerencia: busque en la ayuda de Matlab el algoritmo que emplea la función **roots**).
- A la vista de los resultados anteriores, ¿se producirá la estabilización del número de personas infectadas en el caso propuesto? ¿Podría haberse anticipado este resultado? ¿Por qué? ¿Le parece el modelo propuesto, para los valores de las constantes a , b , c dados, un modelo realista?

Ejercicio 2

Breve introducción matemática

La representación de funciones a través de series de Taylor es una herramienta muy útil en la aproximación local de funciones y, por lo tanto, en la generación de soluciones a través de métodos numéricos. Puesto que se trata de una serie infinita, sólo se podrá emplear un número finito de términos de la serie para aproximar la función. Afortunadamente, habitualmente basta con unos pocos términos iniciales para aproximar la función con la precisión deseada.

El error en la aproximación dependerá de los términos descartados al truncar la serie, por lo que se denomina *error de truncamiento*.

El desarrollo en serie de Taylor de una función $f(x)$ alrededor de un punto x_0 : viene dado por:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)(x - x_0)^2}{2!} + \dots + \frac{f^n(x_0)(x - x_0)^n}{n!}$$

Para realizar estos cálculos, una vez efectuado el truncamiento, el ordenador emplea una aritmética de máquina, representando los números mediante un número finito de cifras, cometiendo así, además, errores de truncamiento o redondeo en la evaluación de la función.

En esta práctica analizaremos el error que surge al aproximar una función de tipo exponencial negativa mediante su serie de Taylor centrada en $x_0 = 0$, en función del número de términos iniciales de la serie que tomemos para su aproximación.

El desarrollo en serie de Taylor de la función $f(x) = e^{-x}$ centrado en $x_0 = 0$ es:

$$e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \dots$$

Aplicación biomédica: Concentración de un fármaco en función del tiempo

Planteamiento del problema

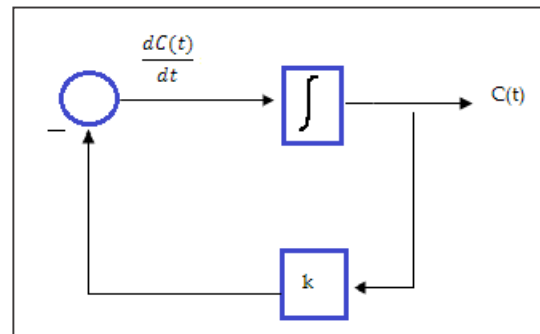
Existen diversos mecanismos a través de los cuales se introducen medicamentos en el cuerpo humano, bien sea por ingesta directa y su posterior absorción por la sangre en el tracto gastrointestinal, por inhalación, aplicación cutánea o subcutánea, inyección intravenosa, etc. La velocidad a la cual estos medicamentos entran en el sistema vascular, el espacio extravascular y posteriormente en el interior celular depende de numerosos factores.

La farmacocinética estudia la distribución de los medicamentos en el organismo y su eliminación, a través de modelos matemáticos que predicen la concentración del fármaco implicado en función del tiempo.

En una reacción química, la velocidad de disminución de la concentración del fármaco es directamente proporcional a la concentración. Esto se puede modelar matemáticamente mediante la siguiente ecuación diferencial:

$$\frac{dC(t)}{dt} = -k C(t).$$

Todos aquellos procesos dinámicos que se puedan modelar en forma de ecuación diferencial implican la existencia en el organismo de un mecanismo de retroalimentación denominado inherente (la retroalimentación no se realiza en forma de materia o información). Se puede comprobar la realimentación negativa existente si expresamos la ecuación anterior en forma de un diagrama de señales.



Para resolver la ecuación diferencial lineal $\frac{dC(t)}{dt} + k C(t) = 0$, generamos en primer lugar su ecuación característica y obtenemos sus raíces:

$$r + k = 0 \Rightarrow r = -k$$

Las soluciones de esta ecuación diferencial lineal serán de la forma:

$$C(t) = \alpha e^{-kt},$$

donde α es una constante a determinar en función de las condiciones iniciales del problema. Si imponemos la condición inicial $C(0) = C_0$, obtenemos $\alpha = C_0$, luego la solución del problema de valor inicial será

$$C(t) = C_0 e^{-kt},$$

donde C_0 es la concentración inicial de plasma con el medicamento en el instante inmediatamente posterior a su introducción y k es la constante de velocidad de eliminación, con unidades SI de s^{-1} , que modela el problema farmacocinético.

Descripción de la práctica

Actividad previa: Obtenga el desarrollo en serie de Taylor de la función $C(t)$. Estos cálculos le ayudarán a entender el código de Matlab que se utilizará en la práctica y le serán útiles a la hora de responder a las cuestiones planteadas.

En el Moodle está disponible el fichero `Pract_error.m`. Descárguelo y léalo con detenimiento intentado comprender el funcionamiento del programa.

A continuación ejecute el código y genere las gráficas, dando el valor 1 a la concentración inicial C_0 y a la constante de velocidad k ; utilice 20 términos de la serie de Taylor ($n = 20$). Añada las gráficas generadas al cuaderno de soluciones.

Modifique el valor del instante t tomando valores desde 0.5 hasta 5 (los que considere oportunos).

- ¿Qué ocurre en las gráficas al aumentar el valor de t ?
- Fijándose en las gráficas obtenidas con tiempos más altos, ¿qué conclusiones puede obtener? ¿es eficiente emplear la serie de Taylor en estos casos?
- Para $t=5$ s, ¿cuántos términos son necesarios para que el error disminuya por debajo de 0.01?

Ahora se analizará el efecto de modificar el resto de constantes.

- Suponiendo $k=1$, $t=1.5$ y $n=20$, obtenga las gráficas que estime necesario para analizar lo que ocurre al variar la concentración inicial C_0 .
- Repita el apartado anterior, pero considerando ahora $C_0=1$, $t=1.5$ y $n=20$ y variando el valor de la constante k .
- Por último, explique el efecto de variar el número de elementos de la serie de Taylor empleados.

Entrega de resultados

Una vez terminados ambos ejercicios cree una carpeta comprimida de nombre `GRxx_P1.zip`, en la que debe incluir el m-file generado en el ejercicio 1 y el documento word de respuestas debidamente cumplimentado (incluyendo las gráficas obtenidas). Finalmente, suba este fichero al Moodle de la asignatura.

Práctica 1

ERRORES

Objetivo de la práctica

Con esta práctica se pretende es que el alumno sea capaz de entender y aplicar, de forma práctica, lo aprendido sobre errores en el tema 2 de la asignatura, de modo que sea consciente de los efectos de la representación interna (en las máquinas) de los números mediante un número finito de dígitos.

Aunque el tema se centra en los errores provocados por la aritmética finita de las máquinas (errores de redondeo y truncamiento debido a la representación finita de los números -*ejercicio 1*), esta práctica incluye también los errores asociados al truncamiento de una serie, en particular la serie de Taylor, a un número finito de términos para permitir realizar los cálculos asociados (*ejercicio 2*). Esto último se da habitualmente en los métodos numéricos, tanto para evaluar funciones como en el propio desarrollo de ciertos métodos numéricos (dando lugar, en tal caso, a errores de método).

De esta manera, el alumno será capaz de analizar de forma práctica los efectos de los distintos tipos de errores que aparecen en el análisis numérico.

Ejercicio 1

Soluciones

- a) Codifique un m-file, de nombre GRxx_E2_1.m, que utilice **format long**, y que calcule las raíces de la ecuación cuadrática en la forma ordinaria y, usando la función **sign**, en la forma recomendada que combina la ordinaria para una raíz y la racionalizada para la otra.

Se escribirá el siguiente código en Matlab:

```
format long;
a=0.0000001;
b=0.123456789;
c=0.0000001;
funcion = [a b c];

%Calculando las raíces de la forma ordinaria.
x1=(-b-sqrt(b^2-4*a*c))/(2*a);
x2=(-b+sqrt(b^2-4*a*c))/(2*a);

%Ahora las calculamos con la fórmula recomendada
q=-(1/2)*(b+sign(b)*sqrt(b^2-4*a*c));
r1=q/a;
r2=c/q;
```

Si nos fijamos en el *workspace*, podemos observar el valor de las variables calculadas:

```
x1 = -1.234567889999190e+06
x2 = -8.099770854030908e-07
r1 = -1.234567889999190e+06
r2 = -8.100000073715315e-07
```

Observamos pequeñas diferencias entre los valores x2 y r2 como consecuencia del empleo de un método u otro.

- b) Calcúlelas también, en el mismo m-file, usando la función **roots** de Matlab.
Empleando la función **roots** el código sería:

```
%Calculamos las raíces con la función 'roots' de Matlab
raices = roots(funcion)
```

El resultado que obtenemos es:

```
r1 = -1.234567889999190 * 1.0e+06
r2 = -0.0000000000000810 * 1.0e+06
```

- c) ¿Qué método usa Matlab? ¿Alguno de los anteriores u otro distinto? (sugerencia: busque en la ayuda de Matlab el algoritmo que emplea la función **roots**).

Matlab utiliza un método distinto en el que para calcular las raíces se calculan los autovalores de una matriz con diagonal inferior en la que los elementos de la primera son $-b/a$ y $-c/a$ (según el algoritmo de la función **roots**).

- d) A la vista de los resultados anteriores, ¿se producirá la estabilización del número de personas infectadas en el caso propuesto? ¿Podría haberse anticipado este resultado? ¿Por qué? ¿Le parece el modelo propuesto, para los valores de las constantes a, b, c dados, un modelo realista?

No se producirá estabilización, pues las raíces (que deberían representar el número de personas infectadas) son todas negativas. La ausencia de estabilización podría haberse anticipado, pues, al ser las constantes a, b, c todas positivas, y N también positivo, la tasa de variación del número de infectados dN/dt será positiva y se producirá un aumento ininterrumpido del número de infectados (de hecho, al ser a, b, c todas positivas la ecuación cuadrática no admitirá soluciones positivas). En este sentido, el modelo propuesto no es realista, pues no considera mecanismos que puedan frenar la propagación de la infección.

Ejercicio 2

El código del algoritmo, disponible en el fichero de Matlab Pract_error.m, es el siguiente:

```
function Pract_error(Co,k,t,n)
% Esta función calcula la aproximación de la serie de Taylor para  $Ce^{-kt}$  desde 2
% a n términos y determina la exactitud en la adición de cada término.
% Variables de entrada
% t es la variable independiente
% n es el máximo número de términos en la serie de Taylor
% Co y k son constantes
% Sólo se consideran los valores positivos de t
%  $c=c_0 * e^{-(k*t)}$ 
err = zeros;
if (t < 0)
t = abs(t);
```

```

end
% Variables adicionales
term = Co; % Primer término en la serie
summation = term; % Término que guarda el valor de la suma
enegativex = Co*exp(-k*t); % Valor verdadero de la función
err(1) = abs(enegativex - summation);
fprintf('\n Series sum Absolute error Percent relative error\n')
for i = 2:n
term = term*(-k*t)/(i-1);
summation = summation + term;

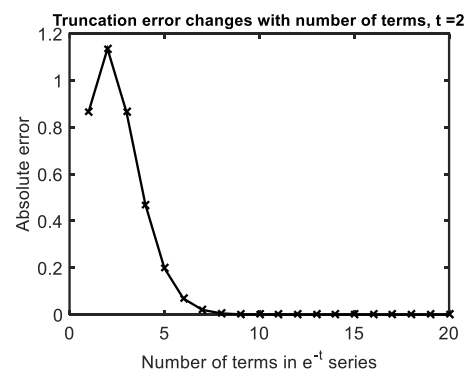
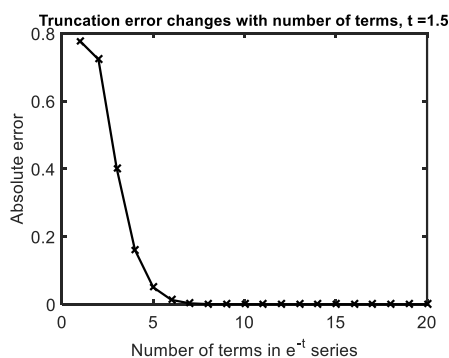
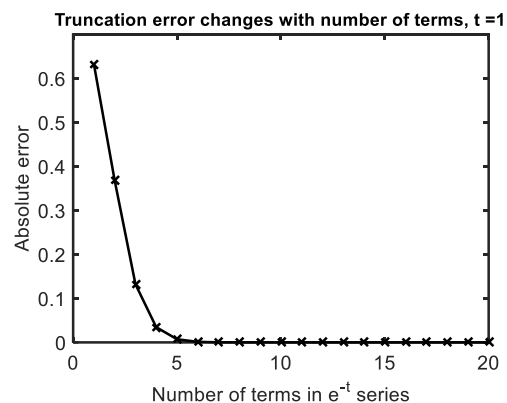
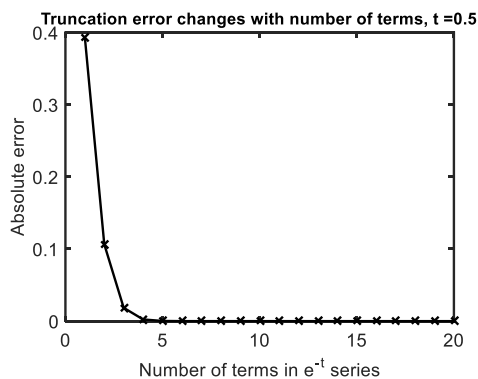
% Error absoluto después de la adición de cada término
err(i) = abs(summation - enegativex);
fprintf('%2d %14.10f %14.10f %18.10f\n',i, summation ,err(i),err(i)/enegativex*100)
end

plot([1:n], err,'k-x','LineWidth',2)
xlabel('Number of terms in e^-t series','FontSize',16)
ylabel('Absolute error','FontSize',16)
title(['Truncation error changes with number of terms, t=',num2str(t)],'FontSize',14)
set(gca,'FontSize',16,'LineWidth',2)

```

Soluciones

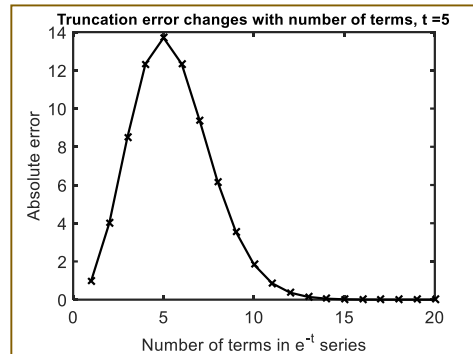
a) ¿Qué ocurre en las gráficas al aumentar el valor de t ?



Como se puede observar, para $t > 1.5$ aproximadamente la serie de Taylor es mucho menos eficiente en aproximar la solución, observándose una especie de campana en la representación del error, que aumenta al aumentar el número de términos de la serie hasta alcanzar un máximo, en tiempos altos, especialmente en $t=5$.

- b) Fijándose en las gráficas obtenidas con tiempos más altos, ¿qué conclusiones puede obtener? ¿Es eficiente emplear la serie de Taylor en estos casos?

Si nos fijamos en el caso $t=5$, se puede observar que el error de la aproximación es peor al usar 2, ..., 10 términos que al usar sólo uno (el primero). Esto se debe a que, cada vez que se incluye un término en la suma de la serie, el numerador (que contiene potencias) aumenta mucho más rápido que el denominador. Hay que tener en cuenta además los errores producidos por el procesamiento inexacto y la representación imprecisa de los resultados, que influye de manera importante en el error.



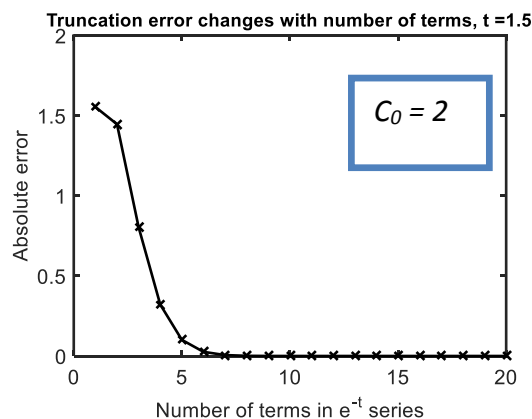
La causa por la que a partir de un cierto número de términos el error vuelve a disminuir es que los términos comienzan a cancelarse (a partir de 15 aproximadamente) y el error de aproximación disminuye considerablemente. Podemos comprobar, por lo tanto, que para tiempos altos es necesaria la inclusión de muchos más términos en la serie de Taylor para aproximar la solución pues, de lo contrario, el error cometido puede ser muy grande.

- c) Para $t=5$ s, ¿cuántos términos son necesarios para que el error disminuya por debajo de 0.01?

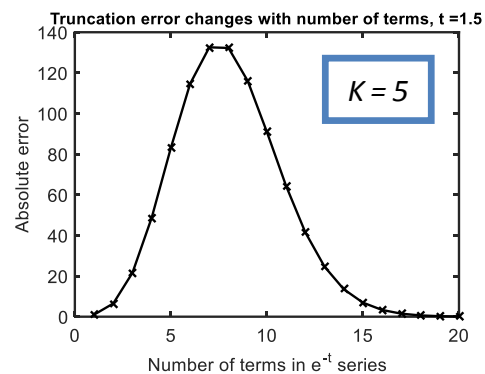
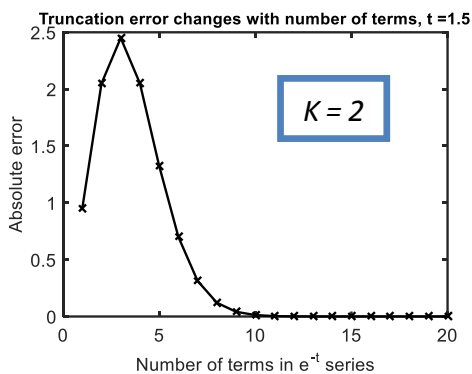
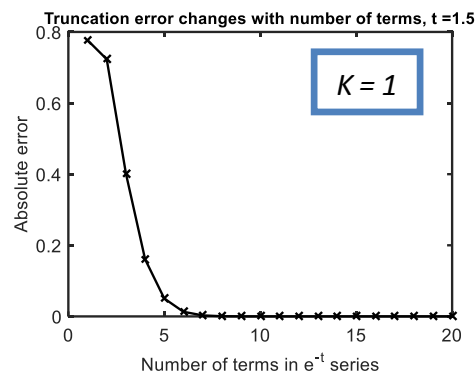
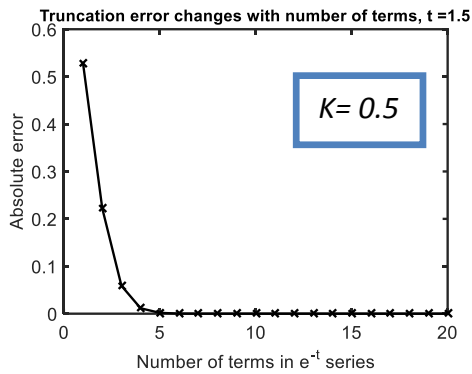
Para el caso de $t=5$, son necesarios aproximadamente 15 términos para disminuir el error por debajo de 0.01.

- d) Suponiendo $k=1$, $t=1.5$ y $n=20$, obtenga las gráficas que estime necesario para analizar lo que ocurre al variar la concentración inicial C_0 .

Si duplicamos la concentración inicial, el error absoluto de la aproximación se duplica de la misma manera: pasamos de un error de 0.8 a un error de 1.6 para un mismo tiempo (1.5 s). Esto es lógico, puesto que se trata de un factor que simplemente multiplica a la serie de Taylor.



- e) Repita el apartado anterior, pero considerando ahora $C_0=1$, $t=1.5$ y $n=20$ y variando el valor de la constante k .



La constante k es la constante de velocidad de eliminación (con unidades SI de s^{-1}). Si nos fijamos en la solución de la ecuación matemática que modela el problema,

$$C(t) = C_0 e^{-kt},$$

podemos observar que esta constante está multiplicando al tiempo, de modo que su efecto será similar al de éste si aumentamos su valor. En efecto, con valores altos de k obtenemos de nuevo una forma de campana en la representación del error con respecto al número de términos incluidos en la suma de la serie.

Hay que recordar que una de las tareas previas consistía en realizar en papel el desarrollo en serie de Taylor de esa exponencial, de manera que rápidamente se puede identificar la causa de este comportamiento observado en las gráficas.

f) Por último, explique el efecto de variar el número de elementos de la serie empleados.

El hecho de incluir un número mayor o menor de términos afecta al error de aproximación, pero siempre ligado al resto de variables. Como hemos podido observar, en función del tiempo y la constante k considerados, el error variaba sustancialmente y el número de términos necesarios para minimizar el error era muy diferente.

Así, el número de términos elegidos para minimizar el error dependerá del resto de parámetros.

Práctica 1

ERRORES

Apellidos, Nombre:

Apellidos, Nombre:

Grupo:

Ejercicio 1

Soluciones

- a) Codifique un m-file, de nombre GRxx_E2_1.m, que utilice **format long**, y que calcule las raíces de la ecuación cuadrática en la forma ordinaria y, usando la función **sign**, en la forma recomendada que combina la ordinaria para una raíz y la racionalizada para la otra. Incluya el resultado de las variables calculadas.
- b) Calcúlelas también, en el mismo m-file, usando la función **roots** de Matlab.
- c) ¿Qué método usa Matlab? ¿Alguno de los anteriores u otro distinto? (sugerencia: busque en la ayuda de Matlab el algoritmo que emplea la función **roots**).
- d) A la vista de los resultados anteriores, ¿se producirá la estabilización del número de personas infectadas en el caso propuesto? ¿Podría haberse anticipado este resultado? ¿Por qué? ¿Le parece el modelo propuesto, para los valores de las constantes a , b , c dados, un modelo realista?

Ejercicio 2

Soluciones

- a) ¿Qué ocurre en las gráficas al aumentar el valor de t ?
- b) Fijándose en las gráficas obtenidas con tiempos más altos, ¿qué conclusiones puede obtener? ¿Es eficiente emplear la serie de Taylor en estos casos?
- c) Para $t=5$ s, ¿cuántos términos son necesarios para que el error disminuya por debajo de 0.01?
- d) Suponiendo $k=1$, $t=1.5$ y $n=20$, obtenga las gráficas que estime necesario para analizar lo que ocurre al variar la concentración inicial C_0 .
- e) Repita el apartado anterior, pero considerando ahora $C_0=1$, $t=1.5$ y $n=20$ y variando el valor de la constante k .
- f) Por último, explique el efecto de variar el número de elementos de la serie empleados.

Espacio para el
profesor

Práctica 2

Sistemas de ecuaciones lineales

Aplicaciones biomédicas

- 🔥 Distribución de tensiones y esfuerzos
- 🔥 Balances de masa metabólicos

Práctica 2

SISTEMAS DE ECUACIONES LINEALES

Objetivos

El objetivo es entender y aplicar, de forma práctica, lo aprendido sobre resolución de sistemas lineales en el tema 3 de la asignatura.

Se aplicarán los distintos tipos de métodos estudiados en el aula (métodos directos, como el de eliminación de Gauss, e iterativos, como los métodos de Jacobi y de Gauss-Seidel) en la resolución de dos problemas biomédicos modelados mediante sistemas de ecuaciones lineales, y se analizarán los resultados obtenidos con cada uno de ellos.

Actividades previas

Antes de la sesión es necesario que lea y comprenda la práctica. De esta forma aprovechará de manera más eficiente las horas de laboratorio.

Notas previas

En esta práctica se utilizarán scripts precodificados de tipo interactivo. Al ejecutarlos, siga las sucesivas instrucciones que el script le dará.

Las respuestas a las cuestiones planteadas, gráficos de soluciones, etc. deben rellenarse y entregarse en el cuaderno de entrega de soluciones disponible en el Moodle de la asignatura.

Ejercicio 1

Aplicación biomédica: distribución de tensiones y esfuerzos

Planteamiento del problema

En el ámbito de la biomecánica de medios continuos es de gran importancia conocer la distribución de tensiones y esfuerzos internos que aparecen en un cuerpo.

Como se estudió en la asignatura de biomecánica de medios continuos, fijado un sistema de referencia ortogonal, esta distribución de tensiones viene dada por el llamado tensor de tensiones de Cauchy, representado por una matriz simétrica

$$\bar{\bar{\sigma}} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{pmatrix}$$

El teorema de Cauchy establece que existe un campo vectorial \bar{t} , llamado campo de tensiones, que varía en cada punto y depende linealmente del vector normal a la superficie \bar{n} , es decir, $\bar{t} = \bar{t}(\bar{x}, \bar{n})$. Esta relación lineal se expresa como $\bar{t} = \bar{\bar{\sigma}}^T \bar{n}$.

Como es simétrica,

$$\bar{t} = \bar{\bar{\sigma}} \bar{n}.$$

Descripción de la práctica

Supongamos que el tensor de Cauchy en un punto viene dado por

$$\bar{\sigma} = \begin{pmatrix} 1 & -2 & 3 \\ -2 & 1 & -1 \\ 3 & -1 & 1 \end{pmatrix}$$

y que el campo de tensiones en ese punto es conocido también:

$$\bar{t} = \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}$$

- Calcule el vector normal a la superficie \bar{n} utilizando el operador barra invertida ($\bar{n} = \bar{\sigma} \backslash \bar{t}$).
- ¿Qué método utiliza Matlab cuando resuelve sistemas lineales de esa forma? Busque en la ayuda de Matlab.

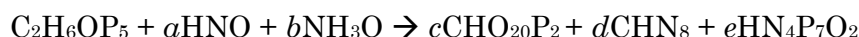
Ejercicio 2

Aplicación biomédica: balances de masa metabólicos

Planteamiento del problema

En el campo de la ingeniería biomédica es esencial el desarrollo de cultivos celulares microbióticos, fundamentalmente en la ingeniería de tejidos, ya que establece los conocimientos básicos sobre el crecimiento y mantenimiento de células ex vivo. Para ello es necesario el uso de sustratos bioquímicos que proporcionen los elementos químicos necesarios para el crecimiento de los microorganismos.

La representación estequiométrica de la fórmula química típica del crecimiento aeróbico de un microorganismo viene dado por



La proporción de moles de CO_2 producidos por mol de O_2 consumido se denomina cociente respiratorio (respiratory quotient, RQ), que se determina empíricamente y se considera un valor conocido.

Para determinar el crecimiento teórico de los microorganismos es necesario calcular los coeficientes estequiométricos de la ecuación: a, b, c, d, e .

Para ello podemos escribir los equilibrios entre masas para cada uno de los elementos individuales:

$$\begin{aligned} \text{C (carbono): } 2 &= c + (RQ)a + d \\ \text{H (hidrógeno): } 6 + 3b + a &= c + d + e \\ \text{O (oxígeno): } 1 + a + b &= 20c + 2e \\ \text{N (nitrógeno): } a + b &= 8d + 4e \\ \text{P (fósforo): } 5 &= 2c + 7e \end{aligned}$$

Nos queda así un sistema de ecuaciones que podemos escribir de forma matricial $\mathbf{Ax} = \mathbf{b}$, con:

$$\mathbf{A} = \begin{pmatrix} RQ & 0 & 1 & 1 & 0 \\ 1 & 3 & -1 & -1 & -1 \\ -1 & -1 & 20 & 0 & 2 \\ -1 & -1 & 0 & 8 & 4 \\ 0 & 0 & -2 & 0 & 7 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 2 \\ -6 \\ 1 \\ 0 \\ 5 \end{pmatrix}.$$

Descripción de la práctica

En el Moodle de la asignatura tiene disponibles tres m-files que contienen los algoritmos necesarios para resolver el sistema anterior mediante los distintos métodos vistos en el aula.

Consideraremos que empíricamente se ha obtenido un cociente respiratorio de 0.2.

Generando las salidas correspondientes en fichero,

- Resuelva mediante el método de eliminación de Gauss, usando `gauss.m`
- Aproxime por el método de Jacobi, usando `jacobi.m`, con una iteración inicial 0 y una tolerancia de 0.01.
- Aproxime por el método de Gauss-Seidel, usando `gausseidel.m`, con una iteración inicial 0 y una tolerancia de 0.01. Asigne un cierto número máximo de iteraciones. Si el método no converge, aumentelo hasta que converja.
- Escriba las soluciones obtenidas con ambos métodos en el cuaderno de entrega de soluciones y compárelas, realizando un breve comentario.
- Verifique si se cumple la condición suficiente de convergencia, para los métodos de Jacobi y Gauss-Seidel, de que la matriz \mathbf{A} sea estrictamente diagonal dominante. Comente el resultado.
- Considerando la solución del método directo de Gauss como exacta, calcule, usando la norma vectorial infinito¹, el error de las aproximaciones de Jacobi y Gauss-Seidel, estimando $\varepsilon_J = \|\mathbf{x}_G - \mathbf{x}_J\|$ y $\varepsilon_{GS} = \|\mathbf{x}_G - \mathbf{x}_{GS}\|$, donde los subíndices G, J, GS corresponden a Gauss, Jacobi y Gauss-Seidel, respectivamente.
- Definiendo la matriz \mathbf{A} y el vector columna \mathbf{b} , calcule directamente la solución usando el mandato de Matlab $\mathbf{x} = \mathbf{A} \backslash \mathbf{b}$.
- ¿Qué conclusiones saca de todos los resultados anteriores? ¿Qué método resulta más adecuado?

Una vez completado el cuaderno de soluciones de la práctica, súbalo al Moodle de la asignatura.

¹ $\|\mathbf{x}\|_\infty = \max_{i=1,\dots,n} |x_i|$

Práctica 2

SISTEMAS DE ECUACIONES LINEALES

Objetivo de la práctica

Entender y aplicar, de manera práctica, lo aprendido sobre resolución de sistemas de ecuaciones lineales en el tema 3 de la asignatura.

Para la resolución del primer ejercicio se aplicarán métodos directos iterativos (eliminación de Gauss) e iterativos (Jacobi y Gauss-Seidel), así como resolución directa con Matlab, con el fin de analizar su eficiencia en la resolución del sistema de ecuaciones planteado.

El segundo ejercicio se centra en aprender a calcular la matriz de iteración de los métodos iterativos expuestos anteriormente y comprender su utilidad en el análisis de la convergencia de dichos métodos.

Ejercicio 1

Soluciones

- a) Calcule el vector normal \bar{n} a la superficie de manera directa ($\bar{n} = \bar{\sigma} \setminus \bar{t}$)

Ejecutando la instrucción en Matlab obtenemos la siguiente solución:

$$\bar{n} = \begin{pmatrix} 3.0000 \\ 20.0000 \\ 13.0000 \end{pmatrix}$$

- b) ¿Qué método utiliza Matlab cuando resuelve sistemas lineales de esa forma? Busque en la ayuda de Matlab.

Emplear la barra '\' en Matlab para resolver sistemas de ecuaciones lineales de la forma $A*x=b$ es equivalente a emplear la función `mldivide(A, b)`. Éste método aprovecha las simetrías que existen en el sistema para resolverlo minimizando el coste computacional. Otro método de Matlab es `linsolve(A, b)`, el cual resuelve el sistema basándose en la factorización triangular de la matriz del sistema.

Ejercicio 2

Soluciones

- a) Resuelva mediante el método de eliminación de Gauss, usando `gauss.m`

En un fichero de texto (.txt) se escribe la matriz de la forma requerida:

```
0.2  0  1  1  0  2
1    3 -1 -1 -1 -6
-1  -1 20  0  2  1
-1  -1  0  8  4  0
0    0 -2  0  7  5
```

Tras ejecutar el código obtenemos lo siguiente:

```
GAUSSIAN ELIMINATION
```

```
The reduced system - output by rows:
```

```

0.20000000 0.00000000 1.00000000 1.00000000 0.00000000 2.00000000
0.00000000 3.00000000 -6.00000000 -6.00000000 -1.00000000 -16.00000000
0.00000000 0.00000000 23.00000000 3.00000000 1.66666667 5.66666667
0.00000000 0.00000000 0.00000000 10.60869565 3.44927536 3.92753623
0.00000000 0.00000000 0.00000000 0.00000000 7.06010929 5.39617486

Has solution vector:
    8.51586687  -4.48490712   0.17511610   0.12171053   0.76431889

with 0 row interchange(s)

```

- b) Aproxime por el método de Jacobi, usando `jacobi.m`, con una iteración inicial 0 y una tolerancia de 0.01.

Establecemos una tolerancia de 0.01, una iteración inicial 0 y un número de iteraciones máximas de 300.

Obtenemos lo siguiente:

```

JACOBI ITERATIVE METHOD FOR LINEAR SYSTEMS

The solution vector is:
    8.52486983 -4.49091715  0.17613299  0.12431696  0.76417661
using 260 iterations
with Tolerance 1.0000000000e-02 in infinity-norm

```

- c) Aproxime por el método de Gauss-Seidel, usando `gausseidel.m`, con una iteración inicial 0 y una tolerancia de 0.01. Asigne un cierto número máximo de iteraciones. Si el método no converge, aumentelo hasta que converja.

Aproximamos ahora por el método Gauss-Seidel. Las condiciones serán las mismas que antes.

```

GAUSS-SEIDEL METHOD FOR LINEAR SYSTEMS

The solution vector is :
    8.51274769 -4.48364306  0.17501843  0.12145406  0.76429098
using 14 iterations
with Tolerance 1.0000000000e-02 in infinity-norm

```

- d) Escriba las soluciones obtenidas con ambos métodos en el cuaderno de entrega de soluciones y compárelas, realizando un breve comentario.

Recopilando las soluciones obtenidas, tenemos:

Gauss:

8.51586687 -4.48490712 0.17511610 0.12171053 0.76431889

Jacobi:

8.52486983 -4.49091715 0.17613299 0.12431696 0.76417661

Gauss-Seidel:

8.51274769 -4.48364306 0.17501843 0.12145406 0.76429098

El método de Gauss es un método exacto, de modo que cabe esperar que, salvo que se produzcan errores de redondeo muy grandes, su solución será la más exacta. Los otros dos métodos, sin embargo, se basan en iteraciones y su resultado

será aproximado. No obstante, para sistemas grandes, los métodos iterativos no sólo son más eficientes sino que incluso sus soluciones pueden llegar a ser más exactas que las obtenidas con métodos directos, debido a la posible acumulación de errores de redondeo en estos últimos. En cuanto a la comparación entre Jacobi y Gauss-Seidel, este último obtuvo la solución, con la precisión deseada (tolerancia de 0.01), en sólo 14 iteraciones frente a las 260 requeridas por el método de Jacobi. El método de Gauss-Seidel resulta, pues, más eficiente.

- e) Verifique si se cumple la condición suficiente de convergencia, para los métodos de Jacobi y Gauss-Seidel, de que la matriz A sea estrictamente diagonal dominante. Comente el resultado.

La matriz A es la siguiente:

$$A = \begin{pmatrix} 0.2 & 0 & 1 & 1 & 0 \\ 1 & 3 & -1 & -1 & -1 \\ -1 & -1 & 20 & 0 & 2 \\ -1 & -1 & 0 & 8 & 4 \\ 0 & 0 & -2 & 0 & 7 \end{pmatrix}$$

Que la matriz sea diagonal dominante implica que para todas las filas, el valor absoluto del elemento de la diagonal de esa fila es estrictamente mayor que la norma del resto de elementos de esa fila. En este caso, esto se cumple para las tercera, cuarta y quinta filas, pero no para las dos primeras, por lo que no es diagonal dominante y, por lo tanto, no se cumple la condición. El hecho de que la condición sea suficiente implica que si se cumple entonces los métodos convergen, pero si no, no se puede asegurar que no vayan a converger, lo cual sucedería si la condición fuera necesaria. De hecho, se ha visto en los apartados anteriores que los métodos sí convergen a la solución del sistema.

- f) Considerando la solución del método directo de Gauss como exacta, calcule, usando la norma vectorial infinito, el error de las aproximaciones de Jacobi y Gauss-Seidel, estimando $\varepsilon_J = \|\mathbf{x}_G - \mathbf{x}_J\|$ y $\varepsilon_{GS} = \|\mathbf{x}_G - \mathbf{x}_{GS}\|$, donde los subíndices G, J, GS corresponden a Gauss, Jacobi y Gauss-Seidel, respectivamente.

Para ello escribimos el siguiente código en Matlab:

```
A= [0.2 0 1 1 0; 1 3 -1 -1 -1; -1 -1 20 0 2; -1 -1 0 8 4; 0 0 -2 0 7]
B=[2; -6; 1; 0; 5]

gauss = [8.51586687 -4.48490712 0.17511610 0.12171053 0.76431889]
jacobi = [8.52486983 -4.49091715 0.17613299 0.12431696 0.76417661]
gaussseid = [8.51274769 -4.48364306 0.17501843 0.12145406 0.76429098]

resta1 = abs(gauss - jacobi)
resta2 = abs(gauss - gaussseid)

eJ = max(resta1)
eGS = max(resta2)
```

Obteniendo los siguientes resultados:

```
resta1 =    0.0090    0.0060    0.0010    0.0026    0.0001
resta2 =    0.0031    0.0013    0.0001    0.0003    0.0000
eJ =    0.0090
eGS =    0.0031
```

Como podemos comprobar, el error absoluto es algo menor aplicando el método de Gauss-Seidel, aunque la diferencia no es excesiva.

- g) Definiendo la matriz **A** y el vector columna **b**, calcule directamente la solución usando el mandato de Matlab **x = A\b**.

```
A= [0.2 0 1 1 0; 1 3 -1 -1 -1; -1 -1 60 0 2; -1 -1 0 8 4;0 0 -2 0 700]
B= [2; -6; 1; 0; 5]
x=A\b
```

Las soluciones que obtenemos son las siguientes:

$$\mathbf{x} = \begin{pmatrix} 8.515866873065015 \\ -4.484907120743034 \\ 0.175116099071207 \\ 0.121710526315789 \\ 0.764318885448916 \end{pmatrix}$$

Como podemos observar, los resultados son los mismos que los obtenidos con el método de Gauss.

Práctica 2

SISTEMAS DE ECUACIONES LINEALES

Apellidos, Nombre:

Apellidos, Nombre:

Grupo:

Ejercicio 1**Soluciones**

- a) Calcule el vector normal a la superficie \bar{n} de manera directa ($\bar{n} = \bar{\sigma} \setminus \bar{t}$).
- b) ¿Qué método utiliza Matlab cuando resuelve sistemas lineales de esa forma? Busque en la ayuda de Matlab.

Ejercicio 2**Soluciones**

- a) Solución obtenida mediante el método de eliminación de Gauss.
- b) Solución obtenida mediante el método de Jacobi, con una iteración inicial 0 y una tolerancia de 0.01.
- c) Solución obtenida mediante el método de Gauss-Seidel, con una iteración inicial 0 y una tolerancia de 0.01.
- d) Compare las tres soluciones obtenidas realizando un breve comentario.
- e) Verifique si se cumple la condición suficiente de convergencia, para los métodos de Jacobi y Gauss-Seidel, de que la matriz \mathbf{A} sea estrictamente diagonal dominante. Comente el resultado.
- f) Considerando la solución del método directo de Gauss como exacta, calcule, usando la norma vectorial infinito, el error de las aproximaciones de Jacobi y Gauss-Seidel, estimando $\varepsilon_J = \|\mathbf{x}_G - \mathbf{x}_J\|$ y $\varepsilon_{GS} = \|\mathbf{x}_G - \mathbf{x}_{GS}\|$, donde los subíndices G, J, GS corresponden a Gauss, Jacobi y Gauss-Seidel, respectivamente.
- g) Definiendo la matriz \mathbf{A} y el vector columna \mathbf{b} , calcule directamente la solución usando el mandato de Matlab $\mathbf{x} = \mathbf{A} \setminus \mathbf{b}$.
- h) ¿Qué conclusiones saca de todos los resultados anteriores? ¿Qué método resulta más adecuado?

**Espacio para el
profesor**

Práctica 3

Ecuaciones no lineales

Aplicaciones biomédicas

- 🔺 Permeabilidad de fluidos en hidrogeles
- 🔺 Deformación plástica en biomateriales

Práctica 3

ECUACIONES NO LINEALES

Objetivos

Existen numerosos problemas en ingeniería biomédica que se pueden modelar con ecuaciones o sistemas no lineales. A diferencia de los sistemas lineales, que pueden resolverse por métodos directos, que son exactos salvo errores de redondeo (aunque también pueden resolverse mediante métodos iterativos), las ecuaciones y sistemas no lineales rara vez pueden resolverse por métodos exactos, siendo necesario recurrir al empleo de métodos numéricos, habitualmente iterativos.

El objetivo de esta práctica es familiarizarse con los métodos presentados en el aula, entender cuál es el procedimiento que siguen para obtener las soluciones, el criterio que usan para detención del algoritmo y ser capaces de determinar el método más eficaz para cada tipo de problema.

Actividades previas

Antes de la sesión es necesario que lea y comprenda la práctica. De esta modo aprovechará de manera más eficiente las horas de laboratorio.

Notas previas

Las respuestas a las cuestiones planteadas, gráficos de soluciones, etc. deben rellenarse y entregarse en el cuaderno de entrega de soluciones disponible en el Moodle de la asignatura.

Atención: Los valores de los parámetros indicados en los ejercicios hay que introducirlos en las unidades dadas, SIN hacer conversión de unidades.

Ejercicio 1

Aplicación biomédica: permeabilidad de fluidos en hidrogeles

Planteamiento del problema

En un medio poroso, como es el caso de los geles de agarosa o la matriz extracelular, es de especial importancia conocer la permeabilidad hidráulica k , que relaciona el gradiente de presiones con la velocidad del fluido en el medio.

En una matriz porosa de fibras, k viene dado por:

$$k = \frac{r_f^2 \varepsilon^3}{20(1-\varepsilon)^2},$$

donde r_f es el radio de la matriz y ε es la porosidad ($0 \leq \varepsilon \leq 1$).

La ecuación anterior se puede reescribir como:

$$f(\varepsilon) = 20k - 40k\varepsilon + 20k\varepsilon^2 - r_f^2 \varepsilon^3 = 0$$

Descripción de la práctica

Para una fibra de radio 10 nm y una permeabilidad hidráulica $k = 0.465 \text{ nm}^2$, determine la porosidad ε ($0 \leq \varepsilon \leq 1$) mediante los siguientes métodos:

- El método de la bisección, usando el m-file `biseccion.m`. Analice también el código para determinar qué criterio de detección se utiliza.
- El método de Regula Falsi, usando el m-file `regFalsi.m`.
- El método de la secante, usando el m-file `secante.m`.
- El método de Newton-Raphson, usando el m-file `newtonRaphson.m`.
- Sabiendo que el valor correcto de ε , con 9 cifras significativas, es $\varepsilon=0.342555712$, calcule el error relativo de cada una de las aproximaciones obtenidas.
- Discuta las ventajas e inconvenientes de los métodos empleados. ¿Cuál de los métodos es más eficaz en este caso? ¿Por qué? Para ilustrar su respuesta cree un m-file que genere una gráfica de la función $f(\varepsilon)$ para valores de ε entre 0.3 y 0.4 (puede usar además la opción `zoom` para ampliar la zona de interés). Inserte la gráfica creada en el cuaderno de resultados.

Para todos ellos especifique una tolerancia de 0.0001, un número máximo de 20 iteraciones (si en algún caso no es suficiente, cambie a 25 el número de iteraciones) y las aproximaciones iniciales 0 (si solo requiere una) ó 0 y 0.5 (si requiere dos), e incluya las **salidas completas** (ε , $f(\varepsilon)$, n° iteraciones y tolerancia, pero NO las iteraciones intermedias) en el fichero de soluciones que tiene que rellenar (puede extraer los datos de salida directamente de la pantalla o a través de un fichero de texto).

Ejercicio 2

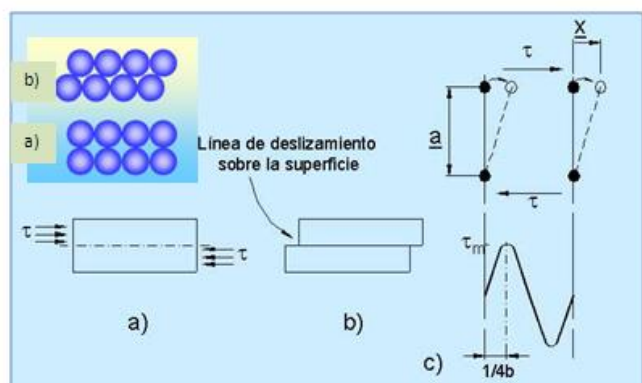
Aplicación biomédica: deformación plástica en biomateriales

Planteamiento del problema

Durante la deformación plástica de un material cristalino se produce un cambio permanente en la forma del mismo. Al aplicar un esfuerzo de cizalladura, los distintos planos paralelos a esta fuerza se deslizan unos sobre otros, siendo el resultado una deformación macroscópica del cristal.

Esta deformación plástica se obtiene suponiendo que el movimiento de todos los átomos del plano que desliza se produce de forma simultánea. Cada vez que se produce un salto atómico (movimiento relativo de los átomos de un plano frente a los de otro plano paralelo), en las posiciones inicial y final (equilibrio estable) y en la intermedia, cuando los átomos de ambos planos están exactamente unos sobre otros (equilibrio inestable), la tensión necesaria para hacer deslizar un plano sobre otro es nula.

Se puede suponer entonces que se necesita una fuerza periódica de cizalladura para mover la fila de átomos del plano superior sobre la fila inferior, y que se puede representar como una función



sinusoidal según vemos en la imagen de la derecha.

Así, para desplazar los átomos una distancia x desde la posición de equilibrio hay que aplicar una tensión de cizalladura, τ , de la forma

$$\tau = \tau_{max} \sin \frac{2\pi x}{b},$$

donde b es la distancia interatómica en el plano de la dirección de deslizamiento (que coincide con la dirección de la tensión de cizalladura).

Descripción de la práctica

Considere que estamos realizando un ensayo con níquel para comprobar sus propiedades en cuanto a deformación plástica a la hora de emplearlo como aleante en el acero inoxidable con objeto de fabricar una prótesis de cadera. Teóricamente se considera un valor de $\tau_{max} = 2600$ MPa y $b = 2.34$ Å, y experimentalmente se ha obtenido un valor de $\tau = 3.1$ MPa. Sabiendo que la solución está en el intervalo $[2,3]$, halle la distancia x que se desplazan los átomos unos sobre otros usando:

- El método de la bisección, usando el m-file bisección.m.
- El método de Regula Falsi, usando el m-file regFalsi.m
- El método de la secante, usando el m-file secante.m
- Cree un m-file que utilice las funciones **fzero** y **optimset** de Matlab para calcular el cero de la función dentro del intervalo considerado, mostrando por pantalla los resultados de las iteraciones intermedias.
- ¿Cuál de los métodos es más eficaz en este caso? Justifique su respuesta.

De nuevo para todos ellos especifique una tolerancia de 0.0001, un número máximo de 20 iteraciones (que puede cambiar a 25 en caso de requerirse) y como aproximaciones iniciales los extremos del intervalo, e incluya las **salidas completas** en el fichero de soluciones (puede extraer los datos de salida directamente por la pantalla o a través de un fichero de texto).

Hay que **subir al Moodle sólo la hoja de soluciones, SIN m-files**.

Práctica 3

ECUACIONES NO LINEALES

Objetivo de la práctica

El objetivo de esta práctica es que el alumno emplee los distintos métodos de resolución de ecuaciones no lineales vistos en la teoría (método de la bisección, método de Regula-Falsi, método de la secante y método de Newton-Raphson) para hallar las raíces de ecuaciones no lineales que aparecen en problemas biomédicos concretos.

Consta de dos ejercicios, en los que se pide utilizar varios de estos métodos para resolver la misma ecuación no lineal con el fin de que el alumno sea capaz de analizar cuál es el método más eficaz para resolver ese problema.

En todos los ejercicios resueltos a continuación se incluyen únicamente las soluciones finales, aunque en el enunciado de la práctica se pidan las salidas completas con todas las iteraciones intermedias.

Ejercicio 1

Soluciones

- a) Método de la bisección. Analice también el código para determinar qué criterio de detección se utiliza.

Obtenemos la siguiente aproximación:

```
Approximate solution P = 0.34259033
with F(P) = -0.00157537
Number of iterations = 14
Tolerance = 1.00000000e-04
```

En la detección del algoritmo se emplea el tercer criterio visto en el aula:

$$|F(P)| < \xi$$

En el código aparece de la siguiente forma:

```
% STEP 4
FP = F(P);
if FLAG == 2
fprintf(OUT, '%3d    %15.8e    %15.7e    %15.8e %15.7e %15.8e\n', I, P, FP, A, FA, B, FB);
end
if abs(FP) < 1.0e-20 | C < TOL
```

- b) Método de Regula-Falsi.

En este caso, 20 iteraciones no son suficientes, por lo que ampliamos el número máximo de iteraciones a 25 tal y como indica el enunciado.

```
Approximate solution P = 0.34235296
with F(P) = 0.00967993
Number of iterations = 22
Tolerance = 1.00000000e-04
```


c) **Método de la secante**

Fijando unas aproximaciones iniciales de 0 y 0.5, y una tolerancia de 0.0001, obtenemos la siguiente aproximación:

Approximate solution P = **0.34255713**
 with F(P) = **-0.00000020**
 Number of iterations = **7**
 Tolerance = **1.00000000e-04**

d) **Método de Newton-Raphson**

Introduciendo una aproximación inicial de 0 y una tolerancia de 0.0001, obtenemos la siguiente aproximación:

Approximate solution = **3.4255712132e-01**
 with F(P) = **-1.3781909836e-08**
 Number of iterations = **5**
 Tolerance = **1.0000000000e-04**

e) **Sabiendo que el valor correcto de ε , con 9 cifras significativas es $\varepsilon=0.342555712$, calcule el error relativo de cada una de las aproximaciones obtenidas.**

Se escriben en un m-file los siguientes datos para obtener los errores relativos de todas las aproximaciones obtenidas:

```
p=0.342555712; %Valor considerado como exacto
pBis=0.34259033;
pSec=0.34255713;
pReg=0.34235296;
pNewt=3.4255712132e-01;
erBis=abs(p-pBis)/abs(p)
erSec=abs(p-pSec)/abs(p)
erReg=abs(p-pReg)/abs(p)
erNewt=abs(p-pNewt)/abs(p)
```

Los errores relativos correspondientes a cada método son en este caso los siguientes:

Método	Error relativo
Bisección	1.0106e-04
Secante	4.1395e-06
Regula Falsi	5.9188e-04
Newton-Raphson	4.1141e-06

f) **Discuta las ventajas e inconvenientes de los métodos empleados. ¿Cuál de los métodos es más eficaz en este caso? ¿Por qué? Para ilustrar su respuesta cree un m-file que genere una gráfica de la función $f(\varepsilon)$ para valores de ε entre 0.3 y**

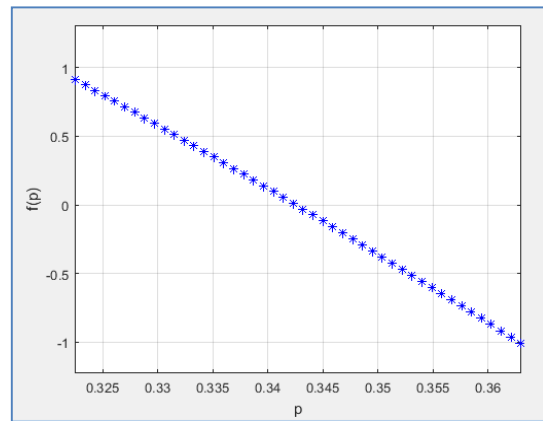
0.4 (puede usar además la opción *zoom* para ampliar la zona de interés). Inserte la gráfica creada en el cuaderno de resultados.

```
x = linspace (0.3, 0.4);
rf = 10;
k = 0.465;
y = 20*k - 40*k*x + 20*k*x.^2 - rf^2*x.^3;
plot(x,y)
```

En cuanto a **eficiencia**, el método de Newton-Raphson es en este caso el más eficiente, puesto que es el que requiere el menor número de iteraciones para obtener la raíz con la precisión/tolerancia especificada.

En lo que respecta a **exactitud**, midiendo ésta en términos de errores relativos, los obtenidos con los métodos de la secante y de Newton-Raphson son mucho más pequeños, del orden de 10^{-6} .

En el caso que nos ocupa el criterio de detención utilizado ($F(p)$ suficientemente pequeño) resulta adecuado, ya que la función es monótona decreciente y, en consecuencia, no se da el caso posible analizado en la teoría de la asignatura en el que $f(p_n)$ sea muy cercano a 0 y sin embargo p_n difiera significativamente de p .



Ahora bien, los métodos de la secante y Newton-Raphson presentan un inconveniente fundamental: resultan muy sensibles a la elección de la(s) aproximación(es) inicial(es), de forma que en general funcionan muy bien si la aproximación inicial está suficientemente próxima al cero buscado, pero, de no ser así, podrían no converger. Frente a ellos, el método de la bisección es el más sencillo, no requiere el empleo de la derivada de la función para hallar la solución (tampoco el de la secante lo requiere) y además siempre converge. Pero, en general, converge muy lentamente (14 iteraciones en nuestro caso).

En resumen, para este caso el método de Newton-Raphson es el más adecuado, aunque hay que tener en cuenta que, en general, la elección del mejor método dependerá de la función y de las aproximaciones iniciales.

Ejercicio 2

Soluciones

a) Método de la bisección:

```
Approximate solution P = 2.34002686
with F(P) = 0.18748654
Number of iterations = 14 Tolerance = 1.00000000e-04
```

b) Método de Regula-Falsi:

```
Approximate solution P = 2.34000000  
with F(P) = 0.00000000  
Number of iterations = 6 Tolerance = 1.00000000e-04
```

c) Método de la secante:

```
Approximate solution P = 2.34000000  
with F(P) = -0.00000000  
Number of iterations = 6  
Tolerance = 1.00000000e-04
```

d) Utilice las funciones **fzero** y **optimset** de Matlab para calcular el cero de la función dentro del intervalo considerado, mostrando por pantalla los resultados de las iteraciones intermedias.

```
opciones = optimset('Display','iter');  
raiz = fzero(@definefuncion,[2,3],opciones);
```

El m-file definefuncion.m sería

```
function y = definefuncion(x)  
y = 2600 * sin(2*pi*x/2.34);
```

Func-count	x	f(x)	Procedure
2	2	-2060.49	initial
3	2.44746	736.758	interpolation
4	2.32961	-75.6505	interpolation
5	2.34058	0.956855	interpolation
6	2.34044	-0.000116748	interpolation
7	2.34044	-0.000116748	interpolation

r2 = 2.3404

e) ¿Cuál de los métodos es más eficaz en este caso? Justifique la respuesta.

En este caso, el método de Regula-Falsi y el de la secante proporcionan ambos una solución exacta con muy pocas iteraciones (únicamente 6). Esto se debe a que en nuestro caso, la función seno es monótona decreciente, y localmente casi lineal, en el entorno del cero, luego no existe una diferencia significativa en el empleo de un método u otro.

Práctica 3

ECUACIONES NO LINEALES

Apellidos, Nombre:

Apellidos, Nombre:

Grupo:

Ejercicio 1**Soluciones**

- a) Método de la bisección. Analice también el código para determinar qué criterio de detección se utiliza.
- b) Método de Regula Falsi..
- c) Método de la secante.
- d) Método de Newton-Raphson.
- e) Sabiendo que el valor correcto de ε , con 9 cifras significativas es $\varepsilon=0.342555712$, calcule el error relativo de cada una de las aproximaciones obtenidas.
- f) Discuta las ventajas e inconvenientes de los métodos empleados. ¿Cuál de los métodos es más eficaz en este caso? ¿Por qué? Para ilustrar su respuesta cree un m-file de nombre GRxx_P3_1.m que genere una gráfica de la función $f(x)$ para valores de x entre 0.3 y 0.4 (puede usar además la opción *zoom* para ampliar la zona de interés). Inserte la gráfica creada en el cuaderno de resultados.

Ejercicio 2**Soluciones**

- a) Método de la bisección.
- b) Método de Regula Falsi.
- c) Método de la secante.
- d) Utilice las funciones **fzero** y **optimset** de Matlab para calcular el cero de la función dentro del intervalo considerado, mostrando por pantalla los resultados de las iteraciones intermedias.
- e) ¿Cuál de los métodos es más eficaz en este caso? Justifique su respuesta.

Recuerde que, para esta práctica, sólo debe subir al Moodle esta hoja de soluciones, SIN añadir los m-files de apoyo usados para generar la gráfica o calcular los ceros con la función **fzero.**

Espacio para el
profesor

Práctica 4

Ajuste por mínimos cuadrados

Aplicaciones biomédicas

- 🔥 Desintegración radiactiva
- 🔥 Curva disociación de la hemoglobina

Práctica 4

AJUSTE POR MÍNIMOS CUADRADOS**Objetivo**

El objetivo de esta práctica es entender y aplicar, de forma práctica, lo aprendido sobre ajuste a una curva de un conjunto de datos discretos empleando el método de mínimos cuadrados.

Actividades previas

Antes de la sesión es necesario que lea y comprenda la práctica. De esta forma aprovechará de manera más eficiente las horas de laboratorio.

Nota previa

Las respuestas a las cuestiones planteadas, gráficos de soluciones, etc. deben rellenarse y entregarse en el cuaderno de entrega de soluciones disponible en el Moodle de la asignatura.

Ejercicio 1**Aplicación biomédica: desintegración radiactiva****Planteamiento del problema**

Muchos fenómenos físicos exhiben una dependencia exponencial con alguna variable, como pueden ser el tiempo, la temperatura o la concentración de sustancias químicas. Éste es el caso de los procesos de desintegración nuclear.

La desintegración de todos los núcleos de una cierta masa no se sucede a intervalos iguales de tiempo sino que obedece a leyes estadísticas. Si N es la cantidad de núcleos radiactivos presentes en la muestra en un instante t , entonces la tasa de cambio de N viene dada por

$$\frac{dN}{dt} = -\lambda N$$

donde λ es la constante de decaimiento, es decir, la probabilidad de que un núcleo decaiga en un instante de tiempo. La radiactividad, por tanto, describe cuántos átomos radiactivos se desintegran por unidad de tiempo.

Resolviendo la ecuación diferencial anterior obtenemos la ley general de desintegración radiactiva, que matemáticamente se expresa como

$$N(t) = N_0 e^{-\lambda t}$$

donde N_0 es el número de átomos presentes para $t=0$. La ecuación muestra un decaimiento exponencial de la radiactividad. Esta ecuación es fundamental durante el empleo de radionúclidos en las imágenes de medicina nuclear.

Aunque la ecuación de diferencias de origen es lineal homogénea, su solución $N(t)$ es de tipo exponencial. Para realizar un ajuste de datos de observación a una función de este tipo, resulta conveniente transformarla de forma que resulte una de tipo lineal. Para linealizar $N(t)$ basta con aplicar logaritmos neperianos a ambos miembros:

$$\log N = \log N_0 - \lambda t$$

Ahora la dependencia de N con t es lineal y se puede emplear una línea recta del tipo $y=ax+b$ para aproximar los datos. Para ello basta con realizar las siguientes sustituciones:

$$y = \log N, \quad x = t, \quad a = -\lambda, \quad b = \log N_0.$$

Si imponemos que la recta y “pase” por el conjunto de puntos (x_i, y_i) , $i = 1, \dots, n$, esto se puede representar de forma matricial como un sistema compatible indeterminado $y=Ac$, donde A es una matriz $n \times 2$ y c es un vector columna:

$$A = \begin{pmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix}, \quad c = \begin{pmatrix} a \\ b \end{pmatrix}.$$

Descripción de la práctica

Supongamos que estamos realizando un estudio de imagen SPECT, que es un tipo de imagen de medicina nuclear, y queremos observar el decaimiento radiactivo que sufre el tecnecio-99 (Tc-99m), una sustancia que se usa como radionúclido en este tipo de imágenes.

Se han obtenido empíricamente los siguientes datos durante los 30 primeros minutos de estudio:

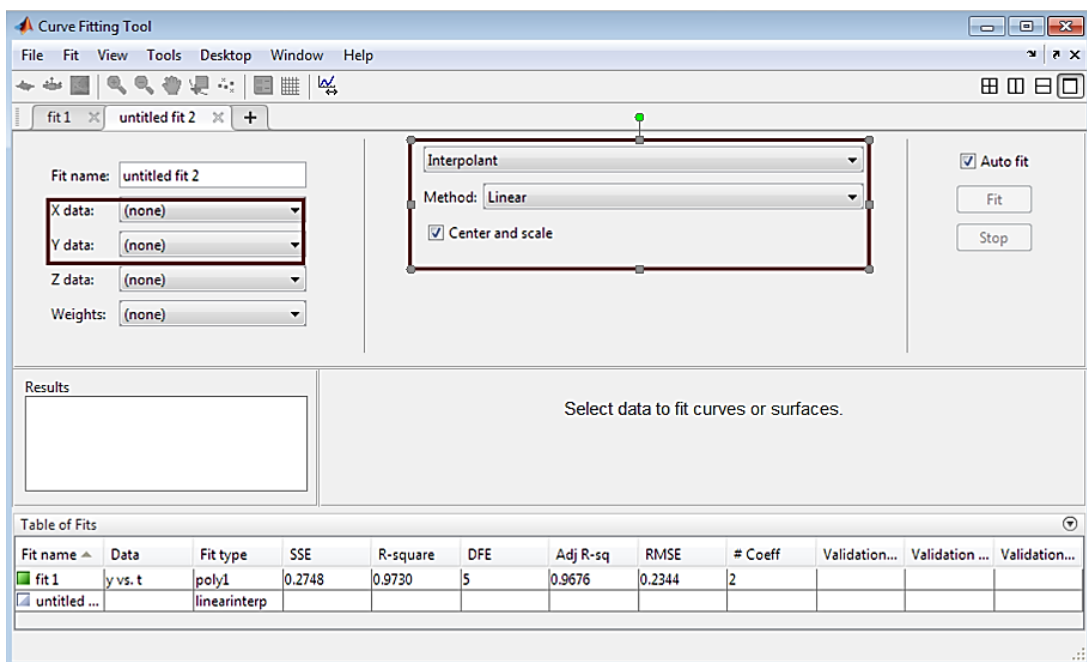
Tiempo (min)	Radiactividad (MBq)
1	$3 \cdot 10^2$
2	$2.80 \cdot 10^2$
6	$1.5 \cdot 10^2$
10	$0.7 \cdot 10^2$
14	$0.45 \cdot 10^2$
20	$0.2 \cdot 10^2$
30	$0.1 \cdot 10^2$

- Se quiere calcular la dosis inicial N_0 y la constante de decaimiento λ . Para ello, cree un script, de nombre GRxx_P4_E1.m, en el que se defina:
 - Un vector temporal t .
 - Un vector que represente los valores de N .
 - Los vectores x e y .
 - La matriz A y el vector columna c . este último se obtiene de la forma $c=(A' A) \backslash (A' y)$ (aquí la prima representa la traspuesta, como en la notación de Matlab –nótese que es una matriz real).
- Dibuje (mediante el mismo script) una gráfica en la que se muestren los valores discretos y su recta de ajuste por mínimos cuadrados. Incluya el gráfico en el cuaderno de resultados.

- c) Matlab incluye una Toolbox™ denominada *Curve Fitting* que permite, entre otras acciones, el ajuste de conjuntos discretos de datos por mínimos cuadrados. Busque en la ayuda de Matlab más información sobre esta herramienta, así como las opciones de ajuste que hay que seleccionar para realizar el ajuste por mínimos cuadrados.

A continuación cargue los datos transformados logarítmicamente en los valores de las variables X e Y de la aplicación y realice el ajuste. Indique los valores que ha obtenido para las constantes a y b de la recta de ajuste, así como el valor de R^2 .

- d) ¿Se puede decir en este caso que la recta de mínimos cuadrados ajusta bien los datos originales?



Ejercicio 2

Aplicación biomédica: curva disociación hemoglobina

Planteamiento del problema

Como se estudió en la asignatura de Fisiología Humana, la hemoglobina es una hemoproteína de la sangre que transporta el oxígeno desde los órganos respiratorios hasta los tejidos, el dióxido de carbono desde los tejidos hasta los pulmones, que lo eliminan, y también participa en la regulación de pH de la sangre. La hemoglobina es una proteína tetrámera, que consta de cuatro cadenas polipeptídicas, dos cadenas α y dos cadenas β , a cada una de las cuales se une un grupo hemo, cuyo átomo de hierro es capaz de unir de forma reversible una molécula de oxígeno. Cuando la hemoglobina tiene unido oxígeno se denomina oxihemoglobina o hemoglobina oxigenada, dando el aspecto rojo característico de la sangre arterial. Cuando pierde el oxígeno, se denomina

hemoglobina reducida, y presenta el color rojo oscuro de la sangre venosa. El aumento de oxigenación de la hemoglobina en los pulmones y la rápida liberación de oxígeno en los tejidos gracias a los efectos del pH y $p\text{CO}_2$ es conocido como *efecto Bohr*. La ecuación que describe la unión entre la hemoglobina y el oxígeno es la siguiente:



Si se aplica la ley de masas al estado de equilibrio químico anterior obtenemos la siguiente ecuación:

$$S = \frac{[\text{Hb}(\text{O}_2)_n]}{[\text{Hb}] + [\text{Hb}(\text{O}_2)_n]} = \frac{(p\text{O}_2)^n}{(p\text{O}_2)^n + P_{50}^n}.$$

Esta ecuación representa la relación entre la concentración de hemoglobina saturada de oxígeno y la presión parcial de oxígeno $p\text{O}_2$ en sangre. P_{50} describe la presión parcial de oxígeno cuando están ocupados la mitad de los sitios de unión de la hemoglobina. A partir de la ecuación se obtiene la conocida curva sigmoidea de disociación de la hemoglobina.

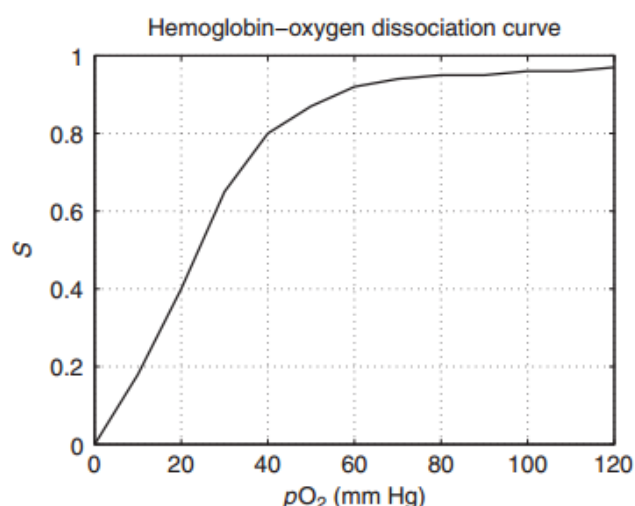


Fig. 4.1 Gráfico de saturación de hemoglobina S como función de la presión parcial de oxígeno en plasma.

Descripción de la práctica

En los últimos años, muchas universidades y laboratorios han estado investigando el empleo de la hemoglobina bovina polimerizada como posible sustituto sanguíneo. Los datos que se obtuvieron en uno de los ensayos se muestran en la tabla siguiente:

$p\text{O}_2$ (mmHg)	S	$p\text{O}_2$ (mmHg)	S
10	0.18 ± 0.01	70	0.94 ± 0.01
20	0.4 ± 0.015	80	0.95 ± 0.02
30	0.65 ± 0.05	90	0.95 ± 0.035
40	0.8 ± 0.034	100	0.96 ± 0.043
50	0.87 ± 0.02	110	0.96 ± 0.067
60	0.92 ± 0.01	120	0.97 ± 0.05

Se quiere determinar los valores que mejor ajusten P_{50} y n en la ecuación anterior para los datos medidos y compararlos con los valores medidos para la hemoglobina

en humanos. El problema es que la ecuación es, de nuevo, no lineal, luego hay que convertirla a lineal reordenando términos primero y a continuación tomando logaritmos en ambos miembros de la ecuación. El resultado es el siguiente:

$$\frac{S}{1-S} = \frac{(pO_2)^n}{P_{50}^n} \Rightarrow \log\left(\frac{S}{1-S}\right) = n \log(pO_2) - n \log(P_{50})$$

- Análogamente a como se hizo en el ejercicio 1, determine para esta ecuación las sustituciones y , x , a y b que deben realizarse para poder escribir la ecuación en la forma $y = ax + b$.
- Calcule los valores de P_{50} y n mediante un script, de nombre GRxx_P4_E2.m, siguiendo los pasos indicados en el ejercicio 1a. Para los valores de la función S , tome los de la tabla SIN los errores asociados (este último se usará en el apartado c).
- Dibuje un gráfico en el que aparezcan los valores discretos y la recta de ajuste por mínimos cuadrados.
- Dibuje otro gráfico en el que se represente la curva de disociación de la hemoglobina bovina a partir de los valores obtenidos de P_{50} y n , usando para ello la ecuación (4.5). Incluya en el gráfico, además, las barras de error correspondientes a los datos discretos usando la función **errorbar**.
- El valor de P_{50} de la hemoglobina en humanos es 26mmHg. ¿Es mayor o menor que el obtenido para la hemoglobina bovina? ¿Qué conclusiones, desde el punto de vista fisiológico, puede extraer de los resultados?

Una vez completado el cuaderno de soluciones de la práctica, comprímalo, junto con los m-files de nombre GRxx_P4_E1.m y de nombre GRxx_P4_E2.m, en un fichero .zip o .rar y súbalo al Moodle de la asignatura.

Práctica 4

AJUSTE POR MÍNIMOS CUADRADOS

Objetivo de la práctica

El objetivo de esta práctica es entender y aplicar, de forma práctica, lo aprendido sobre ajuste a una curva de un conjunto de datos discretos empleando el método de mínimos cuadrados.

Ejercicio 1

Soluciones

- a) Se quiere calcular la dosis inicial N_0 y la constante de decaimiento λ .

Se escribirá el siguiente código en Matlab:

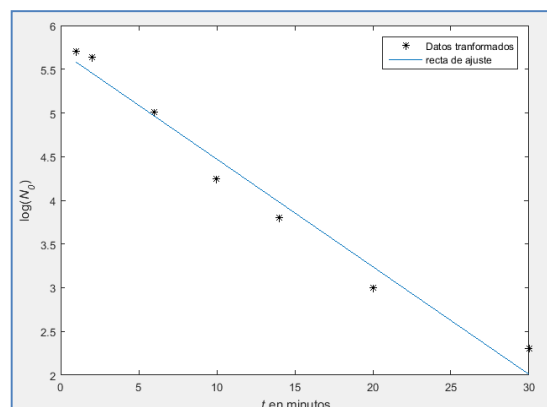
```
t = [1 2 6 10 14 20 30]';
N = ([3 2.8 1.5 0.7 0.45 0.2 0.1]*10^2)';
x = t; y = log(N);
A = [x ones(size(x))];
c = (A'*A)\(A'*y);
lambda = -c(1)
N0 = exp(c(2))
%Dibujamos el resultado
plot(x(1:length(N)),y,'k*',x(1:length(N)),A*c)
xlabel('\it t en minutos')
ylabel('log(\it N_0)')
legend('Datos transformados','recta de ajuste')
```

Obtenemos los siguientes valores para λ y N_0 , respectivamente:

$\lambda = 0.1231$

$N_0 = 299.8015$

- b) Dibuje una gráfica en la que se muestren los valores discretos y su recta de ajuste por mínimos cuadrados. Incluya el gráfico en el cuaderno de resultados.



- c) Ajuste por mínimos cuadrados de los datos transformados empleando la Toolbox™ de Matlab *Curve Fitting*.

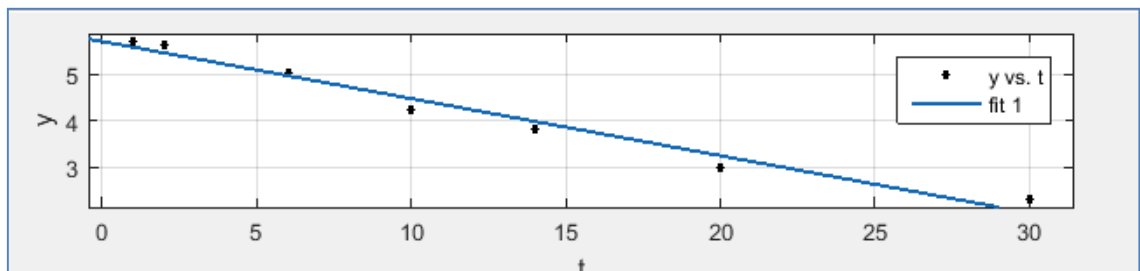
Para que se realice el ajuste por mínimos cuadrados hay que seleccionar la opción **polinomio de grado 1**.

Estos son los resultados que obtenemos:

Linear model Poly1:

$f(x) = p1 \cdot x + p2$
 Coefficients (with 95% confidence bounds):
 $p1 = -0.1231 \quad (-0.1467, -0.09954)$
 $p2 = 5.703 \quad (5.342, 6.064)$

Goodness of fit:
 SSE: 0.2748
 R-square: 0.973
 Adjusted R-square: 0.9676
 RMSE: 0.2344



d) ¿Se puede decir en este caso que la recta de mínimos cuadrados ajusta bien los datos originales?

El valor obtenido de R^2 es de 0.9730, que está muy cercano a 1, por lo que el ajuste es bueno.

Ejercicio 2

Soluciones

a) Análogamente a como se hizo en el ejercicio 1, determine para esta ecuación las sustituciones y , x , a y b que deben realizarse para poder escribir la ecuación de la forma: $y = ax + b$.

$$y = \log\left(\frac{S}{1-S}\right); \quad x = \log(pO_2); \quad a = n; \quad b = -n \log(P_{50})$$

b) Calcule los valores de P_{50} y n mediante un script, de nombre GRxx_P4_E2.m, siguiendo los pasos indicados en el ejercicio 1a.

```

%Sustituciones
pO2 = [10:10:120]';
x = log(pO2);
S = [0.18; 0.40; 0.65; 0.80; 0.87; 0.92; 0.94; 0.95; 0.95; 0.96; 0.96; 0.97];
y = log(S./(1-S));
%Definición de la matriz A y cálculo del vector c
A = [x ones(size(x))];
c = (A'*A)\A'*y;
%Extraemos los valores de las constantes deseadas
n = c(1)
P50 = exp(-c(2)/n) %Transformación para calcular P50
%Calculamos la recta lineal de ajuste
yhat = A*c;
%% Dibujamos los resultados
%Primer gráfico: curva de disociación con los datos transformados
plot(x,y,'k*',x,yhat)
set(gca,'FontSize',16,'LineWidth',2)
xlabel('log(\it{p}O_2)')
ylabel('log(\it{S})/(1-\it{S})')
legend('datos transformados','recta de ajuste')
  
```

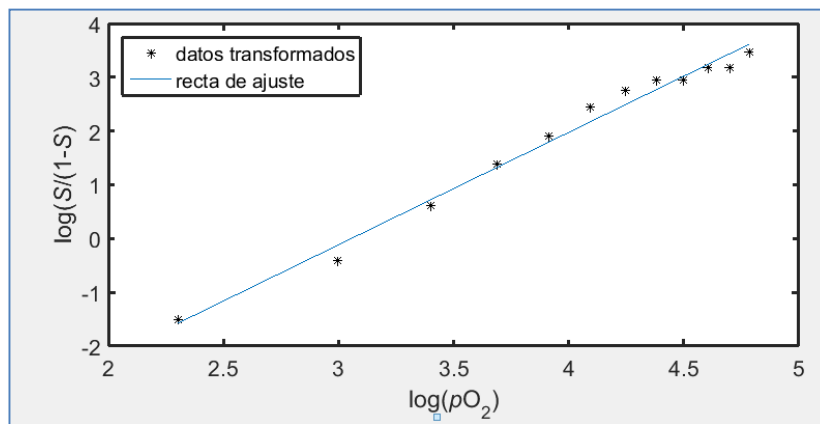
```
%Segundo gráfico: curva de disociación con los datos originales
S_hat = (pO2.^n)./(P50^n+(pO2.^n));
plot(pO2, S, 'k*', pO2, S_hat)
set(gca, 'FontSize', 16, 'LineWidth', 2)
xlabel('\itpO_2 in mmHg')
ylabel('\itS')
legend('datos originales', 'recta de ajuste')
%Dibujamos las barras de error
E = [0.01 0.015 0.05 0.034 0.02 0.01 0.01 0.02 0.035 0.043 0.067 0.05];
errorbar(pO2, S, E, 'rx')
```

Obtenemos los siguientes valores para n y P_{50} :

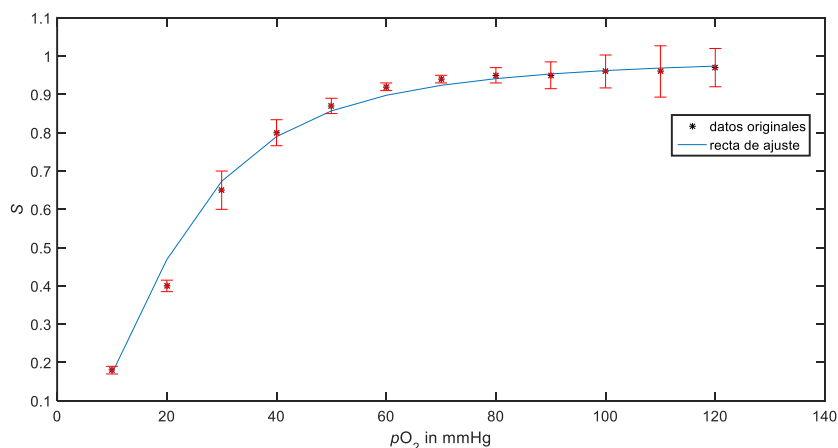
$$n = 2.0906$$

$$P_{50} = 21.2297$$

- c) Dibuje un gráfico en el que aparezcan los valores discretos y la recta de ajuste por mínimos cuadrados.



- d) Dibuje otro gráfico en el que se represente la curva de disociación de la hemoglobina bovina a partir de los valores obtenidos de P_{50} y n , usando para ello la ecuación (4.5). Incluya en el gráfico, además, las barras de error correspondientes a los datos discretos usando la función **errorbar**.



- e) El valor de P_{50} de la hemoglobina en humanos es 26mmHg. ¿Es mayor o menor que el obtenido para la hemoglobina bovina? ¿Qué conclusiones, desde el punto de vista fisiológico, puede extraer de los resultados?

El valor obtenido de P_{50} es 21.2297 mmHg, que es menor que el de la hemoglobina en humanos. Esto indica una mayor afinidad de la hemoglobina bovina por el oxígeno que la humana.

Práctica 4

AJUSTE POR MÍNIMOS CUADRADOS

Apellidos, Nombre:

Apellidos, Nombre:

Grupo:

Ejercicio 1

Soluciones

- a) Cree un m-file, de nombre GRxx_P4_E1.m, siguiendo los pasos indicados en el cuaderno de enunciados.
- b) Dibuje un gráfico en el que aparezcan los valores discretos y la recta de ajuste por mínimos cuadrados. Incluya el gráfico en el cuaderno de resultados.
- c) Busque en la ayuda de Matlab más información sobre esta herramienta, así como las opciones de ajuste que hay que seleccionar para realizar el ajuste por mínimos cuadrados. Realice el ajuste e indique los valores que ha obtenido para las constantes a y b de la recta de ajuste, así como el valor de R^2 .
- d) ¿Se puede decir en este caso que la recta de mínimos cuadrados ajusta bien a los datos originales?

Ejercicio 2

Soluciones

- a) Análogamente a como se hizo en el ejercicio 1, determine para esta ecuación las sustituciones y , x , a y b que deben realizarse para poder escribir la ecuación en la forma $y = ax + b$.
- b) Calcule los valores de P_{50} y n mediante un script, de nombre GRxx_P4_E2.m, siguiendo los pasos indicados en el ejercicio 1a.
- c) Dibuje un gráfico en el que aparezcan los valores discretos y la recta de ajuste por mínimos cuadrados.
- d) Dibuje otro gráfico en el que se represente la curva de disociación de la hemoglobina bovina a partir de los valores obtenidos de P_{50} y n , usando para ello la ecuación (4.5). Incluya en el gráfico, además, las barras de error correspondientes a los datos discretos usando la función **errorbar**.
- e) El valor de P_{50} de la hemoglobina en humanos es 26mmHg. ¿Es mayor o menor que el obtenido para la hemoglobina bovina? ¿Qué conclusiones, desde el punto de vista fisiológico, puede extraer de los resultados?

Recuerde que, para esta práctica, debe subir al Moodle una carpeta comprimida de nombre GRxx_P4 en la que incluya esta hoja de soluciones junto con los dos m-files generados.

Espacio para el
profesor

Práctica 5

Interpolación

Aplicaciones biomédicas

- 🔥 Fermentación de la penicilina
- 🔥 Temperatura corporal de un paciente

Práctica 5

INTERPOLACIÓN**Objetivos**

Los ingenieros deben afrontar la tarea de interpretar y correlacionar observaciones experimentales que suelen venir dadas en forma de datos discretos. Los polinomios de interpolación permiten representar los datos discretos mediante funciones continuas, por lo que son muy útiles. Además, su aplicación más significativa es la aproximación de funciones que son difíciles de integrar o diferenciar, facilitando así la integración y diferenciación (que se verán en temas posteriores). El objetivo de esta práctica es analizar un conjunto discreto de valores (función discreta), generar distintos polinomios de interpolación, detectar los posibles errores que se pueden producir y corregirlos.

Actividades previas

Antes de la sesión es necesario que lea y comprenda la práctica. De esta modo aprovechará de manera más eficiente las horas de laboratorio.

Notas previas

Las respuestas a las cuestiones planteadas, gráficos de soluciones, etc. deben rellenarse y entregarse en el cuaderno de entrega de soluciones disponible en el Moodle de la asignatura. Deberá crear una carpeta comprimida de nombre GRxx_P5 en la que incluya dicho cuaderno de soluciones y todos los m-files generados durante la práctica.

Ejercicio 1**Aplicación biomédica: Fermentación de la penicilina****Planteamiento del problema**

La bencil-penicilina o penicilina G es un antibiótico producido por el *Penicillium chrysogenu*. Se trata de antibióticos del grupo de los betalactámicos, empleados en el tratamiento de infecciones debido a su acción principalmente bactericida. La penicilina es un agente de elección en el tratamiento de infecciones por *Streptococcus pyogenes*, *S. pneumoniae* y *Enterococcus*.

Con el fin de acelerar el crecimiento de cultivos de los hongos y, por lo tanto, la producción de antibióticos de penicilina, los científicos trabajan en procesos de fermentación. En el estudio del crecimiento de los microorganismos en dichos procesos interesa hallar el coeficiente respiratorio (del cual ya se habló en la Práctica 2), que es un indicador de la actividad metabólica del organismo.

Supongamos que experimentalmente se han obtenido los siguientes valores de concentración de penicilina en distintos instantes durante el proceso de fermentación:

Tiempo (días)	Concentración de penicilina (µg/ml)	Tiempo (días)	Concentración de penicilina (µg/ml)
0	5.1765	0.6667	11.5113
0.1333	22.0635	0.8000	17.8462
0.2667	86.2670	0.9333	20.7568
0.4000	47.4483	1.0667	10.4270
0.5333	15.2497	1.2	2.9118

Descripción de la práctica

Ejercicio 1.1

Cree un m-file, de nombre GRxx_P5_E1_1.m, que genere una gráfica, con cuatro subgráficas, similar a la de la figura 6.4 de los apuntes.

Para crear las gráficas, interpole la función discretizada que se obtiene a partir de los datos de la tabla (10 puntos equidistantes entre $x=0$ y $x=1.2$), evaluando el interpolante en 100 puntos en ese mismo intervalo. Limite los valores de y entre -20 y 100 , y ponga los textos (nombres de método) como título. Incluya la gráfica generada en el cuaderno de soluciones.

Ejercicio 1.2

En la subgráfica de splines del ejercicio anterior habrá observado que en el extremo izquierdo el spline presenta una oscilación que no se corresponde con la forma real de la función (los datos de la tabla se han generado mediante la función continua **humps** de Matlab, que no presenta tal oscilación). Esta oscilación del interpolante se debe al uso (por defecto) de condiciones de tipo “not-a-knot”. Para corregir esta situación, y profundizar en nuestro análisis, cree un m-file, de nombre GRxx_P5_E1_2.m, que realice lo siguiente (separe la ejecución de los apartados mediante un mandato pause):

- Dibuje, usando línea continua azul, una gráfica de la función **humps**, con 100 valores de x entre 0 y 1.2, limitando los valores de y entre -20 y 100 . Incluya a continuación un mandato *hold on* para que la siguiente gráfica se superponga a ésta.

Cree ahora una gráfica (única, y mediante un único mandato plot) que represente:

- la función **humps** evaluada en 10 puntos entre $x=0$ y $x=1.2$ (use círculos negros como símbolo);
- la función interpolada mediante splines cúbicos con condiciones de tipo “not-a-knot” (dibujando el spline con línea continua roja, usando 100 puntos entre $x=0$ y $x=1.2$);
- la función interpolada mediante splines cúbicos con condiciones de frontera sujeta, especificando pendiente nula en ambos extremos, pese a no corresponder a la pendiente real de **humps** en los extremos (dibujando el spline con línea continua verde, usando 100 puntos entre $x=0$ y $x=1.2$). Tenga precaución, en este último caso, en la especificación de la función que define los puntos de interpolación y en su utilización posterior en el mandato *plot*.

Genere una leyenda con la información respectiva '*humps*', '*nodos*', '*not-a knot*', '*sujeta*' (nótese que incluye también al primer mandato *plot*, únicamente para

humps). Para esta gráfica múltiple no es necesario especificar nuevamente los límites de ejes, puesto que se mantienen debido a la emisión del *hold on* anterior. Emita finalmente un mandato *hold off* para evitar que se apilen a éstas las gráficas generadas en el siguiente apartado. Incluya la figura resultante de esta ejecución en el cuaderno de soluciones.

b) Dibuje nuevamente **humps**, como en el apartado anterior, seguida de un mandato *hold on*. Cree ahora una gráfica (única) que represente:

b1) la función **humps** evaluada en 17 puntos entre $x=0$ y $x=1.2$ (use círculos como símbolo);

b2) y b3) idénticas a las correspondientes del apartado a.

Genere una leyenda como en el apartado anterior y emita finalmente un mandato *hold off* para evitar el apilamiento de figuras sucesivas. Incluya la figura resultante de esta ejecución en el cuaderno de soluciones.

c) A la vista de los anteriores resultados, ¿qué estrategia recomendaría para lograr una interpolación lo más exacta posible de la función **humps**?

Ejercicio 2

Aplicación biomédica: Temperatura corporal de un paciente

Planteamiento del problema

Por las noches y temprano por las mañanas, la temperatura corporal de un paciente en un hospital aumenta dramáticamente hasta que la enfermera detecta este incremento y le administra la medicación necesaria para bajar la fiebre.

En este ejercicio se tratará de interpretar este cambio de temperatura y cuál es su valor máximo.

La temperatura del paciente se registró por un ordenador en intervalos de una hora y los valores se muestran en la siguiente tabla:

Hora (a.m.)	Temperatura (°C)	Hora (a.m.)	Temperatura (°C)
1	37.2	7	42.0
2	37.5	8	40.1
3	37.7	9	39.2
4	38.5	10	38.4
5	38.7	11	38.1
6	39.2	12	37.9

Descripción de la práctica

a) Cree un m-file, de nombre GRxx_P5_E2.m, que genere un interpolante de tipo spline de los 12 valores equiespaciados recogidos en la tabla, usando condiciones de frontera por defecto. Dibuje los 12 puntos utilizados usando círculos como símbolo. Dibuje usando línea continua el spline generado, utilizando para la representación gráfica 31 valores equiespaciados entre $x=1$ y $x=12$, limitando los valores de y entre 37 y 43. Notará que la línea continua no pasa por el nodo de mayor valor de la función discreta (esta discrepancia puede hacerse más evidente

añadiendo un mandato *axis* (*[6 8 41.5 42.5]*)). Incluya la gráfica generada con la opción *axis* en el cuaderno de soluciones.

- b) ¿Se debe el desacuerdo a las condiciones en la frontera? ¿O hay un fallo en la rutina de la función **spline**? ¿O hay otra razón para el desacuerdo? Implemente una corrección para este problema (separe ambos apartados mediante un mandato *pause*). Incluya la gráfica generada (con el problema corregido) en el cuaderno de soluciones. (Pista: sin cambiar los puntos de interpolación, haga la representación gráfica usando 12, 23 ó 45 puntos. ¿Por qué en estos casos la línea continua pasa por el máximo?)
- c) Una vez solucionado el problema, determine la temperatura máxima que alcanza el paciente.

Práctica 5

INTERPOLACIÓN

Objetivo de la práctica

El objetivo de esta práctica es que el alumno sea capaz de analizar un conjunto discreto de valores (función discreta) y generar distintos polinomios de interpolación. Además, debe ser capaz de detectar los posibles errores que se pueden producir, establecer su origen y corregirlos.

Ejercicio 1.1

Soluciones

- a) Crear un m-file que genere una gráfica, con cuatro subgráficas, similar a la de la figura 6.4, a partir de los datos proporcionados

Un código en Matlab posible para crear las gráficas es el siguiente:

```
%% Script Practica 4 ejercicio 1
x= linspace(0,1.2,10); % para insertar los 10 valores
equiespaciados
y= [5.1765, 22.0635, 86.2670, 47.4483,15.2497,11.5113,17.8462,
20.7568,10.4270,2.9118];
xhat = linspace(0,1.2,100); % Interpolamos en 100 puntos

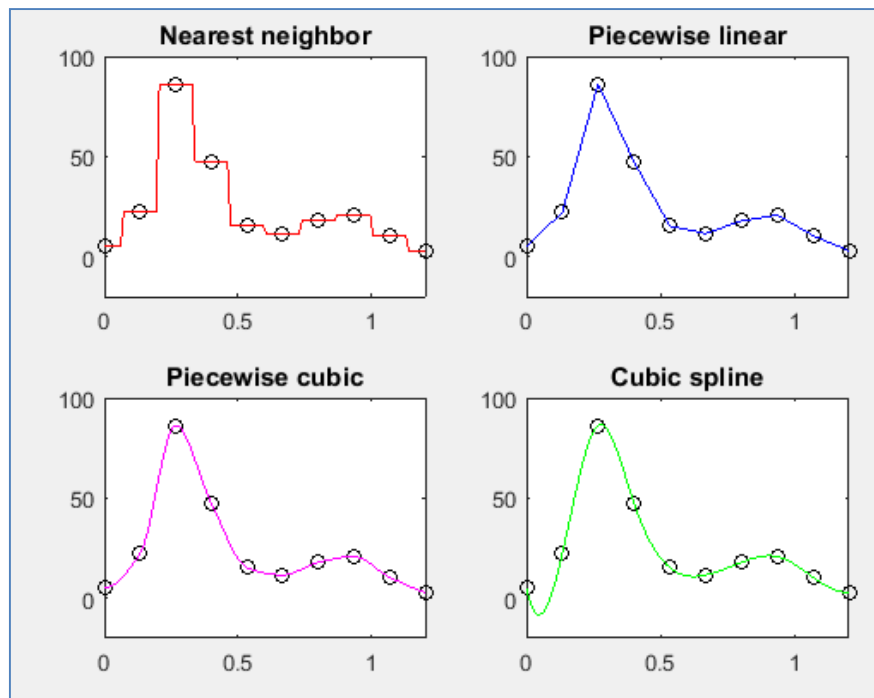
% Grafica del cuadrante 1
subplot(2,2,1)
yhat = interp1(x,y,xhat,'nearest');
plot(x,y,'ko',xhat,yhat,'r-');
axis([0 1.2 -20 100]); title('Nearest neighbor');

% Grafica del cuadrante 2
subplot(2,2,2)
yhat = interp1(x,y,xhat,'linear');
plot(x,y,'ko',xhat,yhat,'b-');
axis([0 1.2 -20 100]); title('Piecewise linear');

% Grafica del cuadrante 3
subplot(2,2,3)
yhat = interp1(x,y,xhat,'cubic');
plot(x,y,'ko',xhat,yhat,'m-');
axis([0 1.2 -20 100]); title('Piecewise cubic');

% Grafica del cuadrante 4
subplot(2,2,4)
yhat = interp1(x,y,xhat,'spline');
plot(x,y,'ko',xhat,yhat,'g-');
axis([0 1.2 -20 100]); title('Cubic spline');
```

Obtenemos las siguientes gráficas:



Ejercicio 1.2

Apartado a)

El siguiente código de Matlab

```
% Script Practica 4 ejercicio 1, parte 2
```

```
%Apartado a)
```

```
x= linspace(0,1.2);
y= humps(x);
plot(x,y,'b-'); axis([0 1.2 -20 100]);
hold on;
```

```
%Primera funcion, a1
```

```
x1=linspace(0,1.2,10);
y1=humps(x1)
plot(x1,y1,'ko');
```

```
%Segunda funcion, a2
```

```
xhat = x;
yhat = interp1(x1,y1,xhat,'spline');
plot(xhat,yhat,'r-')
```

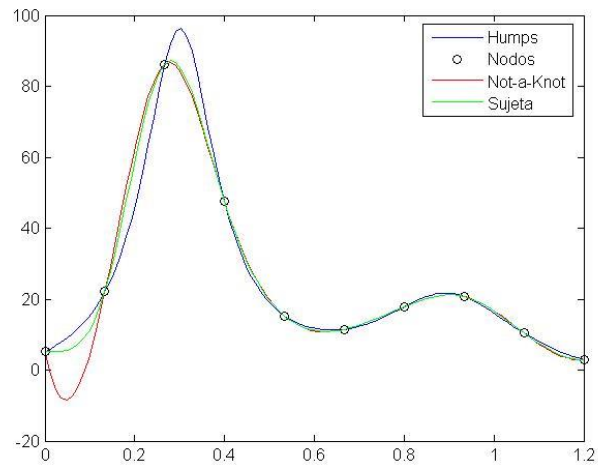
```
%Tercera funcion, a3
```

```
xhat = x;
cs=spline(x1, [0 y1 0]);
plot(xhat,ppval(cs,xhat),'g-')
hold off
```

```
%Creamos la leyenda de la grafica
```

```
legend('Humps','Nodos','Not-a-Knot','Sujeta')
```

genera la gráfica



Apartado b)

El siguiente código de Matlab

```
%% Script Practica 4 ejercicio 1, parte 2
```

```
%Apartado b)
```

```
x= linspace(0,1.2);
y= humps(x);
plot(x,y,'b-'); axis([0 1.2 -20 100]);
hold on;
```

```
%Primera funcion, a1
```

```
x1=linspace(0,1.2,17);
y1=humps(x1);
plot(x1,y1,'ko');
```

```
%Segunda funcion, a2
```

```
xhat = x;
yhat = interp1(x1,y1,xhat,'spline');
plot(xhat,yhat,'r-')
```

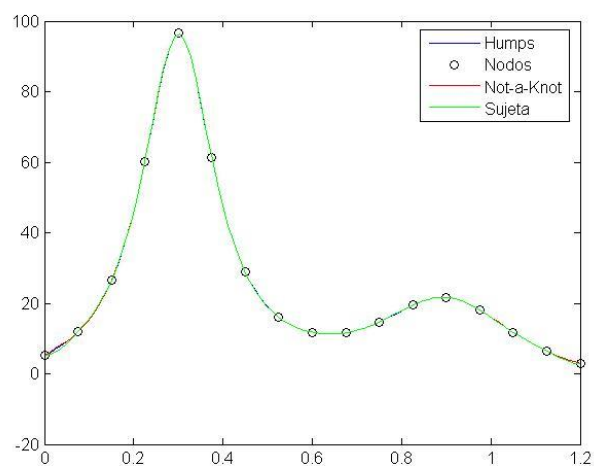
```
%Tercera funcion, a3
```

```
xhat = x;
cs=spline(x1, [0 y1 0]);
plot(xhat,ppval(cs,xhat),'g-')
hold off
```

```
%Creamos la leyenda de la grafica
```

```
legend('Humps','Nodos','Not-a-Knot','Sujeta')
```

Genera la gráfica:



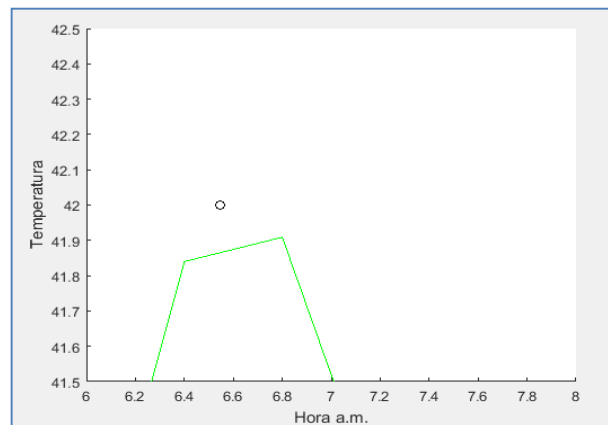
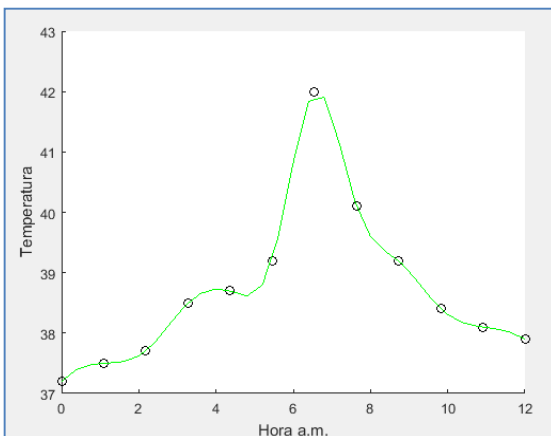
- c) A la vista de los anteriores resultados, ¿qué estrategia recomendaría para lograr una interpolación lo más exacta posible de la función humps?

En general, especificar condiciones de frontera sujeta es lo más recomendable cuando se conocen los valores de la derivada en los extremos. De no ser así, convendrá escoger una interpolación con un mayor número de nodos, porque no puede garantizarse a priori si el uso de condiciones de tipo *not-a-knot* producirá un mejor resultado que, por ejemplo, especificar condiciones de frontera sujeta con derivadas nulas en los extremos.

Ejercicio 2

- a) Cree un m-file, de nombre GRxx_P5_E2.m, que genere un interpolante de tipo spline de los 12 valores equiespaciados recogidos en la tabla, usando condiciones de frontera por defecto. Dibuje los 12 puntos utilizados usando círculos como símbolo. Dibuje usando línea continua el spline generado, utilizando para la representación gráfica 31 valores equiespaciados entre $x=1$ y $x=12$, limitando los valores de y entre 37 y 43. Notará que la línea continua no pasa por el nodo de mayor valor de la función discreta (esta discrepancia puede hacerse más evidente añadiendo un mandato *axis* ([6 8 41.5 42.5])). Incluya la gráfica generada con la opción *axis* en el cuaderno de soluciones.

```
% Apartado a)
x= linspace(1,12,12);
y=[37.2,37.5,37.7,38.5,38.7,39.2,42.0,40.1,39.2,38.4,38.1,37.9];
xhat = linspace(1,12,31);
yhat = interp1(x,y,xhat,'spline');
hold on
plot(x,y,'ko')
plot(xhat,yhat,'g-'); axis([1 12 37 43]);
xlabel('Hora a.m. '); ylabel('Temperatura');
pause
```

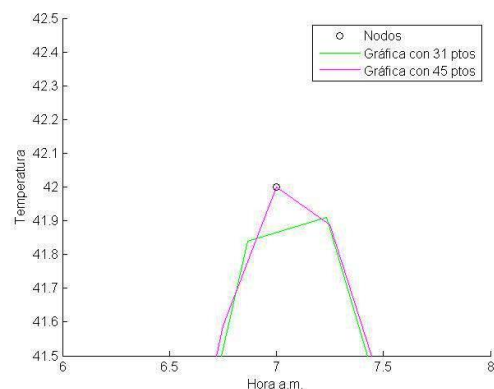
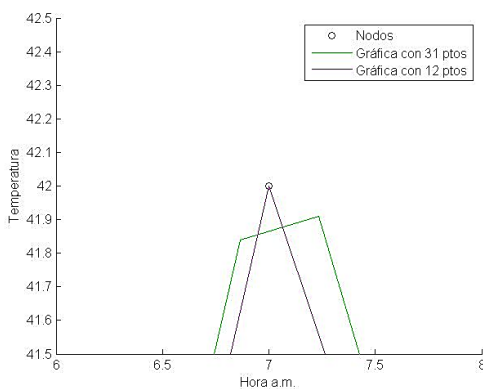


- b) ¿Se debe el desacuerdo a las condiciones en la frontera? ¿O hay un fallo en la rutina de la función spline? ¿O hay otra razón para el desacuerdo? Implemente una corrección para este problema (separe ambos apartados mediante un mandato *pause*). Incluya la gráfica generada (con el problema corregido) en el cuaderno de soluciones. (Pista: sin cambiar los puntos de interpolación, haga la

representación gráfica usando 12, 23 ó 45 puntos. ¿Por qué en estos casos la línea continua pasa por el máximo?)

Las condiciones de frontera no son la causa porque el fallo no se produce en los extremos. Tampoco se debe a un fallo en la función *spline*, sino que el problema está en el número de puntos que hemos tomado en la representación gráfica, que es tal que ninguno de los puntos cae cerca del máximo del spline de interpolación. Sin embargo, si tomamos 12, 23 (12+11), 45 (23+22), ... puntos entonces la línea continua pasa por el máximo. Como norma general, si tomamos un elevado número de puntos para la representación este problema no se dará (o será imperceptible).

```
x= linspace(1,12,12);  
y= [37.2,37.5,37.7,38.5,38.7,39.2,42.0,40.1,39.2,38.4,38.1,37.9];  
xhat = linspace(1,12,12); %Probar también con 23, 45, ...  
yhat = interp1(x,y,xhat,'spline');  
hold on  
plot(xhat,yhat,'m-'); axis([6 8 41.5 42.5]);  
xlabel('Hora a.m. '); ylabel('Temperatura');  
legend('Nodos', 'Gráfica con 31 ptos', 'Gráfica con 12 ptos')
```



- c) Una vez solucionado el problema, determine la máxima temperatura que alcanza el paciente.

Si incluimos en el código el mandato: $\text{maxT}=\text{max}(\text{yhat})$, obtenemos el siguiente resultado: $\text{maxT} = 42.0193$.

Práctica 5

INTERPOLACIÓN

Apellidos, Nombre:

Apellidos, Nombre:

Grupo:

Ejercicio 1.1

- a) Incluya la gráfica, con 4 subgráficas, generada a partir de los datos proporcionados.

Ejercicio 1.2

- a) Incluya la figura generada.
b) Incluya la figura generada.
c) A la vista de los anteriores resultados, ¿qué estrategia recomendaría para lograr una interpolación lo más exacta posible de la función humps?

Ejercicio 2

- a) Incluya la gráfica generada.
b) ¿Se debe el desacuerdo a las condiciones en la frontera? ¿O hay un fallo en la rutina de la función spline? ¿O hay otra razón para el desacuerdo? Una vez corregido el problema, incluya la gráfica generada.
c) Una vez solucionado el problema, determine la temperatura máxima que alcanza el paciente.

Recuerde que, para esta práctica, debe subir al Moodle una carpeta comprimida de nombre GRxx_P5 en la que incluya esta hoja de soluciones junto con los tres m-files generados.

Espacio para el profesor

Práctica 6

Integración numérica

Aplicación biomédica

🚒 Administración de solución salina por vía intravenosa

Práctica 6

INTEGRACIÓN NUMÉRICA

Objetivos

El objetivo de esta práctica es familiarizarse con los métodos de integración numérica basados en las fórmulas de Newton-Cotes (reglas del trapecio y Simpson) y comprender sus limitaciones.

Actividades previas

Antes de la sesión es necesario que lea y comprenda la práctica. De esta modo aprovechará de manera más eficiente las horas de laboratorio.

Nota previa

Las respuestas a las cuestiones planteadas, gráficos de soluciones, etc. deben rellenarse y entregarse en el cuaderno de entrega de soluciones disponible en el Moodle de la asignatura.

Ejercicio 1

Aplicación biomédica: Administración de solución salina por vía intravenosa

Planteamiento del problema

Durante una intervención quirúrgica, los pacientes suelen requerir la administración de líquidos por vía intravenosa para mantener una presión arterial adecuada. Consideremos una bolsa cilíndrica (Figura 1) que contiene la solución salina de NaCl que se suministra al paciente. Por simplificación, se considerará que el líquido fluye por efecto de la gravedad por un agujero situado en la parte inferior de la bolsa y se introduce posteriormente en el flujo sanguíneo del paciente. La única resistencia presente al movimiento de la solución es la resistencia del tubo, la cual determina el tiempo que tarda la bolsa en vaciarse.

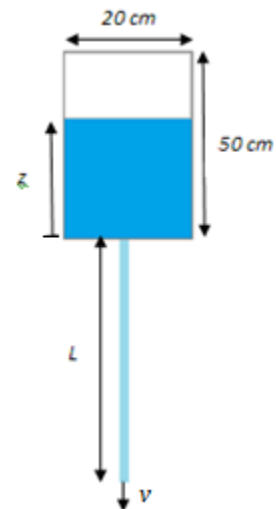


Figura 1

Para calcularlo se planteará un equilibrio cinemático calculado mediante la ecuación de Bernoulli, pero teniendo en cuenta la fricción en el tubo: la energía a la altura del fluido en la bolsa será igual a la energía del líquido saliendo del tubo menos la pérdida por fricción a lo largo del mismo. La altura del fluido en la bolsa con respecto al paciente es, a la vista de la figura, $(z + L)$ cm.

La presión en el interior del tubo debido a la fricción con las paredes viene dada por la ecuación de Hagen-Poiseuille:

$$\Delta P = \frac{8QL\mu}{\pi R_{tubo}^4} = \frac{32vL\mu}{d^2}$$

donde Q es el flujo (caudal) del fluido, d es el diámetro interno del tubo, ΔP es la caída de presión entre los dos extremos, μ es la viscosidad dinámica, L la longitud del tubo a lo largo del eje z y v es la velocidad del fluido a la salida del tubo.

La ecuación de Bernoulli en este problema es la siguiente:

$$g(z + L) = \frac{v^2}{2} + \frac{\Delta P}{\rho} = \frac{v^2}{2} + \frac{32vL\mu}{d^2\rho}$$

donde $\Delta P/\rho$ es el término de pérdida de cinética debido a la fricción.

Despejando u , y considerando $\alpha = \frac{64L\mu}{d^2\rho}$, resolvemos la ecuación de segundo grado:

$$v = \frac{-\alpha + \sqrt{\alpha^2 + 8g(z+L)}}{2}$$

Como el flujo de fluido en la bolsa es igual al del tubo:

$$-\rho\pi R_{bolsa}^2 \frac{dz}{dt} = \rho\pi \frac{d^2}{4} v$$

Reordenando y sustituyendo Q obtenemos finalmente:

$$dt = -4 \frac{R_{bolsa}^2}{d^2 v} dz = - \frac{8R_{bolsa}^2}{(-\alpha + \sqrt{\alpha^2 + 8g(z+L)})d^2} dz$$

Descripción de la práctica

Supongamos que $L = 91.44$ cm, $d = 0.1$ cm, $R_{bolsa} = 10$ cm, la viscosidad de la solución salina $\mu = 0.01$ Poise, la densidad $\rho = 1$ g/cm³, y la gravedad $g = 981$ cm/s². Queremos calcular el tiempo que tarda la bolsa en vaciarse un 90% de su volumen.

La integral que hemos de resolver será:

$$t = \int_a^b - \frac{8R_{bolsa}^2}{(-\alpha + \sqrt{\alpha^2 + 8g(z+L)})d^2} dz = \frac{8R_{bolsa}^2}{d^2} \int_a^b - \frac{1}{(-\alpha + \sqrt{\alpha^2 + 8g(z+L)})} dz$$

a) ¿Qué intervalo de integración $[a,b]$ se debe considerar para resolver el problema?

A continuación, resuelva la integral usando el método del trapecio y el de Simpson. Para ello siga los siguientes pasos:

- Cree una función, de nombre GRxx_P6_b.m, en la que defina los valores de todos los parámetros proporcionados y la función a integrar (el integrando).
- Usando el m-file trapecio.m, resuelva la integral definida de la última ecuación (sin incluir el factor $\frac{8R_{bolsa}^2}{d^2}$), tomando los límites de integración que haya considerado necesarios en el apartado a. Haga el cálculo para un número de subintervalos $n=5$, $n=10$ y $n=20$.
- Repita el cálculo completo del apartado anterior empleando esta vez el algoritmo del m-file Simpson.m.
- ¿Cuánto tiempo tarda la bolsa en vaciarse un 90% en cada caso (trapecio y Simpson)? Utilice para ello únicamente la solución para $n = 20$. Si se quisiera que

el flujo de la solución salina fuera más lento y así el tiempo que tarda la bolsa en vaciarse fuera mayor, ¿qué parámetros modificaría?

- f) Suponiendo que el valor exacto de la integral es 0.574948166362027, ¿qué método proporciona una mejor aproximación?

Para terminar, establezca la opción **format long** y calcule numéricamente el valor de la integral de la función dada mediante:

- g) La función **quad** de Matlab.
h) La función **quadl** de Matlab.

Escriba en el documento Word de entrega de resultados los valores obtenidos con ambas funciones.

- i) ¿Cuál de todos los métodos anteriores proporciona una mejor aproximación de la solución exacta? ¿Qué limitaciones puede decirse que presentan los dos primeros métodos (trapecio y Simpson)?

Entrega de resultados

Una vez terminado el ejercicio, cree una carpeta comprimida de nombre GRxx_P6.zip, en la que debe incluir el m-file generado en el ejercicio y el documento Word de respuestas debidamente cumplimentado. Finalmente, suba este fichero al Moodle de la asignatura.

Práctica 6

INTEGRACIÓN NUMÉRICA

Objetivo de la práctica

El objetivo de esta práctica es familiarizarse con los métodos de integración numérica basados en las fórmulas de Newton-Cotes (reglas del trapecio y Simpson) y comprender sus limitaciones, así como familiarizarse con las funciones **quad** y **quadl** de Matlab.

Ejercicio 1

Soluciones

- a) ¿Qué intervalo de integración $[a,b]$ se debe considerar para resolver el problema?

Buscamos que se vacíe el 90%, o lo que es lo mismo, que quede un 10% del volumen. Puesto que la superficie no varía, bastará con considerar que la altura final sea un 10% de la inicial, que son 50 cm, de modo que el intervalo de integración será entre 50 y 5.

- b) Cree una función, de nombre GRxx_P6_b.m, en la que defina los valores de todos los parámetros proporcionados y la función a integrar.

En el m-file correspondiente se escribe la función:

```
function f = GRxx_P6_b(z)
% This function calculates the integrand function of the IV drip
% problem.
% Parameters
L = 91.44; % cm
mu = 0.01; % Poise
d = 0.1; % cm
rho = 1; % g/cm^3
alpha = 64*L*mu/d^2/rho;
g = 981; % cm/s^2
f = 1./(-alpha + sqrt(alpha^2 + 8*g*(z + L)));
```

Hemos escrito la función con signo positivo suponiendo que intercambiaremos los límites de integración, utilizando $[5,50]$ en lugar de $[50,5]$.

- c) Usando el m-file trapecio.m, resuelva la integral definida de la última ecuación (sin incluir el factor $\frac{8R_{bolsa}^2}{d^2}$), tomando los límites de integración que haya considerado necesarios en el apartado a. Haga el cálculo para un número de subintervalos $n=5$, $n=10$ y $n=20$.

Escribimos en la línea de comandos:
 trapecio(@GRxx_P6_b, 5, 50, n), siendo n
 el número de subintervalos requerido.

n	Valor integral (z)
5	0.575526596539867
10	0.575092911633913
20	0.574984361326113

- d) Repita el cálculo completo del apartado anterior empleando esta vez el algoritmo del m-file Simpson.m.

Escribimos en la línea de comandos:
Simpson(@GRxx_P6_b, 5, 50, n), siendo n el número de subintervalos requerido.

n	Valor integral (z)
5	0.574953819750762
10	0.574948349998595
20	0.574948177890180

- e) ¿Cuánto tiempo tarda la bolsa en vaciarse un 90% en cada caso (trapecio y Simpson)? Utilice para ello únicamente la solución para $n = 20$. Si se quisiera que el flujo de la solución salina fuera más lento y así el tiempo que tarda la bolsa en vaciarse fuera mayor, ¿qué parámetros modificaría?

El tiempo viene dado por

$$t = \frac{8R_{bolsa}^2}{d^2} z$$

Método	Valor integral (z)	Tiempo (s)	Tiempo (horas)
Trapecio	0.574984361326113	45998,74890608906	12.77743
Simpson	0.574948177890180	45995,8542312144	12.77662

Si se aumentara el radio de la bolsa o se disminuyera el radio del tubo, el tiempo aumentaría en la misma proporción.

- f) Suponiendo que el valor exacto de la integral es 0.574948166362027, ¿qué método proporciona una mejor aproximación?

En el caso $n=20$, el método de Simpson aproxima mejor la solución exacta que el método del trapecio.

- g) La función **quad** de Matlab.

- h) La función **quadl** de Matlab.

Para ello, creamos el siguiente m-file:

```
format long
Inumquad = quad(@GRxx_P6_b,5,50)
Inumquadl = quadl(@GRxx_P6_b,5,50)
```

que al ejecutarse proporciona, vía la ventana de mandatos, los siguientes valores:

Inumquad = 0.574948169175127

Inumquadl = 0.574948166373875

- i) ¿Cuál de todos los métodos anteriores proporciona una mejor aproximación de la solución exacta? ¿Qué limitaciones puede decirse que posean los dos primeros métodos (trapecio y Simpson)?

La mejor aproximación la proporcionan las dos funciones de Matlab, y en concreto **quadl**, porque además de incorporar un criterio de detención mejorado, está basado en una regla de interpolación de Guass-Labbato y dos extensiones de Kronrod que hacen más eficientes las iteraciones. **quad**, por su parte, proporciona una buena aproximación ya que está basado en el método de Simpson con un criterio de detención mejorado. Si en el propio método de Simpson se incluyera un número mucho mayor de subintervalos la solución sería mucho más próxima a la exacta, pero acarrearía un mayor coste computacional.

Práctica 6

INTEGRACIÓN NUMÉRICA

Apellidos, Nombre:

Apellidos, Nombre:

Grupo:

Ejercicio 1**Soluciones**

- a) ¿Qué intervalo de integración $[a,b]$ se debe considerar para resolver el problema?
- b) Cree una función, de nombre GRxx_P6_b.m, en la que defina los valores de todos los parámetros proporcionados y la función a integrar.
- c) Usando el m-file trapecio.m, resuelva la integral definida de la última ecuación (sin incluir el factor $\frac{8R_{bolsa}^2}{d^2}$), tomando los límites de integración que haya considerado necesarios en el apartado a. Haga el cálculo para un número de subintervalos $n=5$, $n=10$ y $n=20$.
- d) Repita el cálculo completo del apartado anterior empleando esta vez el algoritmo del m-file Simpson.m.
- e) ¿Cuánto tiempo tarda la bolsa en vaciarse un 90% en cada caso (trapecio y Simpson)? Utilice para ello únicamente la solución para $n=20$. Si se quisiera que el flujo de la solución salina fuera más lento y así el tiempo que tarda la bolsa en vaciarse fuera mayor, ¿qué parámetros modificaría?
- f) Suponiendo que el valor exacto de la integral es 0.574948166362027, ¿qué método proporciona una mejor aproximación?
- g) Incluya el resultado obtenido usando la función **quad** de Matlab.
- h) Incluya el resultado obtenido usando la función **quadl** de Matlab.
- i) ¿Cuál de todos los métodos anteriores proporciona una mejor aproximación de la solución exacta? ¿Qué limitaciones puede decirse que posean los dos primeros métodos (trapecio y Simpson)?

**Espacio para el
profesor**

Práctica 7

Integración numérica de problemas de valor inicial

Aplicaciones biomédicas

- 🚩 Unión monovalente de ligando a receptores de membrana celulares
- 🚩 Infección viral

Práctica 7

**INTEGRACIÓN NUMÉRICA DE PROBLEMAS
DE VALOR INICIAL****Objetivos**

Todos los procesos dinámicos que ocurren en el organismo, como las reacciones enzimáticas o la absorción de fármacos, se modelan matemáticamente mediante ecuaciones diferenciales. La integración numérica de problemas de valor inicial es, por tanto, fundamental en el estudio biomédico. En esta práctica empleará algunos de los métodos vistos en el aula para resolver los problemas planteados y discutir los resultados.

Actividades previas

Antes de la sesión **es necesario que** lea y comprenda la práctica. De esta modo aprovechará de manera más eficiente las horas de laboratorio y podrá terminarla dentro del horario previsto.

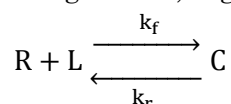
Nota previa

Las respuestas a las cuestiones planteadas, gráficos de soluciones, etc. deben rellenarse y entregarse en el cuaderno de entrega de soluciones disponible en el Moodle de la asignatura.

Ejercicio 1**Aplicación biomédica: Unión monovalente del ligando a los receptores de membrana celulares****Planteamiento del problema**

En términos muy generales se puede definir un ligando como una molécula capaz de ser reconocida por otra provocando una respuesta biológica. El reconocimiento de ligandos por receptores de membrana es un evento clave en la señalización celular. La unión del ligando a su receptor generalmente provoca un cambio conformacional del receptor que puede favorecer la apertura de un canal iónico (como ocurre en la sinapsis), la síntesis de un segundo mensajero al favorecer una reacción enzimática o suponer el inicio de una cascada de señalización intracelular. El hecho de que la unión sea monovalente implica que existe un solo sitio de unión para el ligando.

Consideremos un ligando monovalente L que se une reversiblemente a un receptor R para formar un complejo receptor-ligando C , según la reacción



donde C representa la concentración del complejo receptor (R)-ligando (L) y k_f y k_r son las velocidades de las reacciones directa (forward) e inversa (reverse), respectivamente. Si suponemos que la concentración del ligando se mantiene

constate, L_0 , se puede escribir la siguiente ecuación diferencial que gobierna la dinámica del complejo:

$$\frac{dC}{dt} = K_f[R_T - C]L_0 - K_r C$$

Desarrollo de la práctica

Se realizará la integración de la ecuación diferencial anterior a través de distintos métodos. En primer lugar, abra los m-files de Matlab proporcionados `eulerforwardmethod.m`, `midpointmethod.m` y `RK4method.m`, que contienen algoritmos que se corresponden con el método de Euler básico, un método de Runge-Kutta de 2 etapas y un método de Runge-Kutta de 4 etapas. Observe en los códigos los parámetros de entrada de cada una de las funciones. Como puede comprobar, el primer parámetro de entrada es el nombre del archivo donde está definida la función de interés.

- Cree un m-file de función, de nombre `GRxx_P7_E1a.m`, y defina en él la ecuación diferencial planteada anteriormente y los siguientes valores de las constantes: $k_f=7*10^2$, $k_r=0.6$, $R_T=5*10^3$, $L_0=10^{-6}$.
- Cree un m-file, de tipo script, de nombre `GRxx_P7_E1b.m`, y escriba en él el código necesario para llamar a las funciones anteriores. Para ello:
 - En primer lugar establezca **format long**.
 - Fije a continuación los valores de los parámetros de entrada: concentración de C inicial (y_0) de $5*10^{-8}$, tiempo máximo de 10 s y una longitud de paso de 0.2 s.
 - Seguidamente, llame a las tres funciones almacenando el resultado en variables del tipo $[t, y]$ (matriz de dos componentes).
 - Dibuje los resultados obtenidos en una única gráfica y con un único mandato **plot**. Incorpore un título, nombre en los ejes y una leyenda. Incluya la imagen resultante en el cuaderno de entrega.
- ¿Cuál es la concentración final (último valor almacenado en el vector) en cada método? (Ayuda: los resultados quedan almacenados en el vector y)
- Ahora modifique el valor de h a 0.05. ¿Qué diferencias observa en la gráfica y en los valores de las concentraciones finales?
- Si el resultado exacto de la concentración final es 5.8121934843, ¿qué conclusiones puede extraer? ¿Cuál es el método que mejor aproxima la solución exacta en este caso?

Ejercicio 2

Aplicación biomédica: Infección viral

Planteamiento del problema

Cuando una partícula vírica entra en un organismo, comienza atacando los linfocitos para emplear su maquinaria genética y así producir más virus dentro de la célula. Posteriormente, el genoma vírico programa a la célula infectada para sufrir apoptosis celular y, a continuación, los viriones producidos se introducen en el flujo sanguíneo y son capaces de atacar nuevas células repitiendo el proceso y produciendo una diseminación de la infección. La dinámica de la infección viral y su replicación en plasma se puede modelar como un sistema de ecuaciones diferenciales.

$$\frac{dC^*}{dt} = kVC - \delta C^*$$

$$\frac{dV}{dt} = N\delta C^* - cV$$

donde C es la concentración de las células diana, C^* es la concentración de células infectadas, V es la concentración de virus en plasma, k determina la velocidad a la cual las células diana se infectan, δ es la velocidad a la que se pierden células del sistema inmune por apoptosis o lisis, c es la velocidad a la cual se eliminan partículas infecciosas del flujo sanguíneo y N es el número de viriones producidos por la célula infectada.

En un estudio clínico se probó la eficacia de un antiviral administrándolo a un grupo de infectados. Se pudo comprobar que, una vez que se absorbió el tratamiento, los nuevos virus producidos perdían su capacidad de infectar nuevas células, dejando de ser partículas infecciosas, pero no afectaba ni al proceso de entrada en las células diana ni a la producción de virus en las células ya infectadas.

La dinámica de cómo el antiviral reduce la concentración de partículas virales en plasma se puede modelar, de nuevo, mediante un sistema de ecuaciones diferenciales de primer orden:

$$\begin{aligned}\frac{dC^*}{dt} &= kV^*C - \delta C^* \\ \frac{dV^*}{dt} &= -cV^* \\ \frac{dV_N}{dt} &= N\delta C^* - cV_N\end{aligned}$$

siendo V^* la concentración de virus infecciosos en plasma y V_N la concentración de partículas no virales en plasma. Un estudio experimental proporcionó las siguientes estimaciones de los parámetros y concentraciones iniciales C , C^* , V^* y V_N :

$\delta = 0.5/\text{día}$ $c = 3/\text{día}$ $k = 2 \times 10^{-4} \text{ } \mu\text{l}/\text{día}/\text{virión}$ $N = 60 \text{ viriones por célula}$	$V^*(t=0) = 100/\mu\text{l}$ $V_N(t=0) = 0/\mu\text{l}$ $C(t=0) = 250 \text{ células no infectadas}/\mu\text{l}$ $C^*(t=0) = 10 \text{ células infectadas}/\mu\text{l}$
-----------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Descripción de la práctica

Para resolver el sistema de ecuaciones diferenciales anterior, que rige la dinámica de la infección vírica después del tratamiento, renombramos las variables como sigue:

$$y_1 = C^* \qquad y_2 = V^* \qquad y_3 = V_N$$

Introduciendo también los valores concretos de los parámetros, las ecuaciones quedan así:

$$\begin{aligned}\frac{dy_1}{dt} &= 0.05y_2 - 0.5y_1 \\ \frac{dy_2}{dt} &= -3y_2 \\ \frac{dy_3}{dt} &= 30y_1 - 3y_3\end{aligned}$$

Para resolver este sistema usaremos el algoritmo RK4methodvectorized.m.

- a) Cree en primer lugar un m-file de función de nombre GRxx_P7_E2a.m y defina en él las ecuaciones diferenciales planteadas anteriormente y los valores de las constantes correspondientes. Para ello, considere la función como un vector en el cual cada componente sea una de las ecuaciones diferenciales.

- b) Cree un m-file de tipo script de nombre GRxx_P7_E2b.m y escriba en él el código necesario para llamar a las funciones anteriores. Para ello:
- Primero establezca los valores de las variables de entrada: concentraciones iniciales de C^* , V^* y V_N , tiempo máximo de 5 días y un paso de 0.1. (Guarde los tres parámetros de entrada como componentes de un vector y0)
 - A continuación llame a la función RK4methodvectorized, almacenando el resultado en variables del tipo [t, y].
 - Dibuje tres subgráficas en las que represente, en función del tiempo, C^* , V^* y V_N . Incorpore un título y los nombres de los ejes en cada subgráfica.
- c) Calcule el tiempo que se tarda en reducir a un 50% el número de viriones (Ayuda: piense cómo se almacenan los resultados y utilice un bucle con una condición apropiada). Incluya esta parte del código a continuación del escrito en el apartado anterior. Indique en el cuaderno de respuestas el tiempo calculado.
- d) Interprete las gráficas obtenidas, desde un punto de vista biológico.

Entrega de resultados

Una vez terminados ambos ejercicios, cree una carpeta comprimida de nombre GRxx_P7.zip, en la que debe incluir todos los m-files generados y el documento Word de respuestas debidamente cumplimentado (incluyendo las gráficas obtenidas). Finalmente, suba este fichero al Moodle de la asignatura.

Práctica 7

INTEGRACIÓN NUMÉRICA DE PROBLEMAS DE VALOR INICIAL

Objetivo de la práctica

Con esta práctica se pretende que el alumno no sólo aplique directamente los algoritmos de Matlab proporcionados para resolver las ecuaciones diferenciales planteadas en los dos problemas, si no que sea capaz de escribir el código necesario para solucionar los ejercicios y así adquirir más soltura empleando Matlab. Además, se busca que sea capaz de interpretar desde el punto de vista biológico los resultados obtenidos.

Ejercicio 1

Soluciones

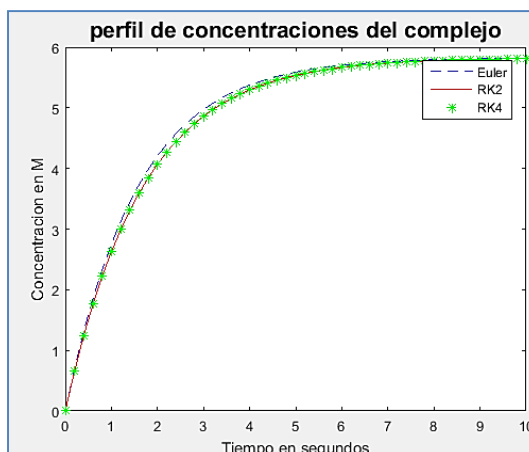
- f) Cree en primer lugar un m-file de función, de nombre GRxx_P7_E1a.m, y defina en él la ecuación diferencial planteada anteriormente y los siguientes valores de las constantes: $k_f=7 \cdot 10^2$, $k_r=0.6$, $R_t=5 \cdot 10^3$, $L_0=10^{-6}$.

```
function f = GRxx_P7_E1a(t,y)
kf = 7*10.^2;
kr = 0.6;
Rt = 5*10.^3;
L0 = 10.^-6;
f=[kf*(Rt-y)*L0-kr*y];
```

- g) Cree un m-file, de tipo script, de nombre GRxx_P7_E1b.m, y escriba en él el código necesario para llamar a las funciones anteriores.

```
% Práctica 7, Ejercicio 1: Unión ligando-receptor
format long
% Parámetros
y0 = 5*10.^-8;
tf = 10; % s
h = 0.2; % s
% Llamadas a las funciones
[t1, y1] = eulerforwardmethod(@GRxx_P7_E1a, tf, y0, h);
[t2, y2] = midpointmethod(@GRxx_P7_E1a, tf, y0, h);
[t3, y3] = RK4method(@GRxx_P7_E1a, tf, y0, h);
% Dibujar resultados
clf;
plot(t1,y1,'k-',t2,y2,'r-',t3,y3,'b*')
title('perfil de concentraciones del complejo','FontSize',16)
xlabel('Tiempo en segundos')
ylabel('Concentracion en M')
legend('Euler','RK2','RK4')
```

Obtenemos la siguiente gráfica:



- h) ¿Cuál es la concentración final (último valor almacenado en el vector) en cada método? (Ayuda: los resultados quedan almacenados en el vector y)

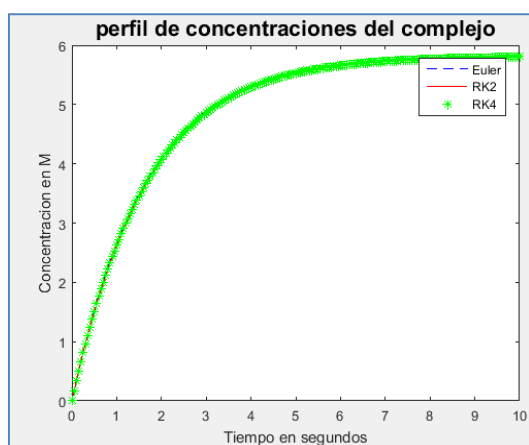
Puesto que, como se indica en el enunciado, los resultados de cada método se guardan, en este caso, en los vectores y1, y2 e y3, simplemente basta con acceder al último componente de cada vector. Añadimos para ello el siguiente fragmento de código:

```
max1 = y1 (length (y1) -1)
max2 = y2 (length (y2) -1)
max3 = y3 (length (y3) -1)
```

Los valores obtenidos son los siguientes:

Euler básico	Runge-Kutta 2	Runge-Kutta 4
5.815528541436646	5.810110515379517	5.810362934943298

- i) Ahora modifique el valor de h a 0.05. ¿Qué diferencias observa en la gráfica y en los valores de las concentraciones finales?



Se puede comprobar, tanto con la gráfica como tomando únicamente los últimos valores, que con un paso más pequeño las soluciones de los tres métodos son mucho más parecidas entre sí y se aproximan más a la solución exacta.

h	Euler básico	Runge-Kutta 2	Runge-Kutta 4
0.2	5.815528541436646	5.810110515379517	5.810362934943298
0.05	5.813050208198564	5.811743035201595	5.811756624641377

- j) Si el resultado exacto de la concentración final es 5.8121934843, ¿qué conclusiones puede extraer? ¿Cuál es el método que mejor aproxima la solución exacta en este caso?

Como se ha determinado en el ejercicio anterior, una disminución del paso mejora la aproximación ya que supone una disminución del error de discretización para cada método. Por ello se trabajará con los últimos valores obtenidos ($h=0.05$). Calculando los errores relativos con respecto al valor exacto:

Método	Error relativo
Euler básico	1.474011319268959e-04
Runge-Kutta 2	-7.750070599357838e-05
Runge-Kutta 4	-7.516261456248999e-05

Los dos métodos de Runge-Kutta se aproximan bien, pero el de cuarto orden es más exacto.

Ejercicio 2

Soluciones

- a) Cree en primer lugar un m-file de función de nombre GRxx_P7_E2a.m y defina en él las ecuaciones diferenciales planteadas anteriormente y los valores de las constantes correspondientes. Para ello considere la función como un vector en el cual cada componente es una de las ecuaciones diferenciales.

```
function f = GRxx_P7_E2a(t, y)
% Constantes
k = 2e-4; % uL/día/viriones
N = 60; % viriones/celula
delta = 0.5; % /día
c = 3.0; % /día
T = 250; % células no infectadas/uL
f = [k*T*y(2)-delta*y(1); -c*y(2); N*delta*y(1)-c*y(3)];
```

- b) Cree un m-file de tipo script de nombre GRxx_P7_E2b.m y escriba en él el código necesario para llamar a las funciones anteriores.

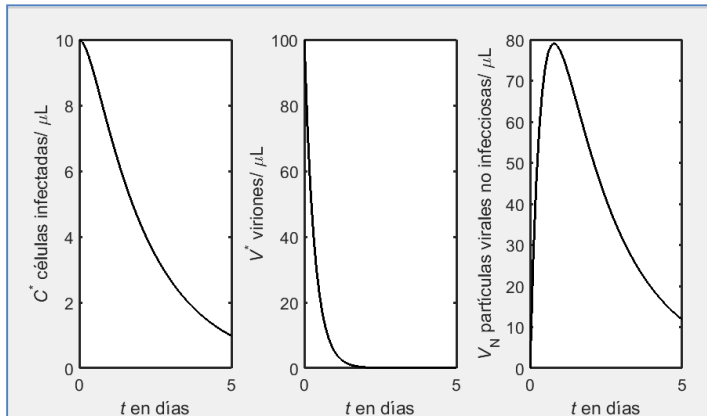
```
% Práctica 7, Ejercicio 2: Infección viral
clear all
% Variables
y0(1) = 10; % initial value of T*
y0(2) = 100; % initial value of V1
y0(3) = 0; % initial value of Vx
tf = 5; % interval size (day)
h = 0.1; % step size (day)
% LLamamos a la función
[t, y] = RK4methodvectorized(@GRxx_P7_E2a, tf, y0, h);
% Gráficas de los resultados
figure;
subplot(1,3,1)
plot(t,y(1,:), 'k-', 'LineWidth', 2)
set(gca, 'FontSize', 16, 'LineWidth', 2)
xlabel('\itt en días')
ylabel('\itC)^* células infectadas/ \muL')
subplot(1,3,2)
```



```

plot(t,y(2,:), 'k-', 'LineWidth', 2)
set(gca, 'FontSize', 16, 'LineWidth', 2)
xlabel('{\itt} en días')
ylabel('{\itV}^* viriones/ \mu L')
subplot(1,3,3)
plot(t,y(3,:), 'k-', 'LineWidth', 2)
set(gca, 'FontSize', 16, 'LineWidth', 2)
xlabel('{\itt} en días')
ylabel('{\itV}_N partículas virales no infecciosas/ \mu L')

```



c) Calcule el tiempo que se tarda en reducir a un 50% el número de viriones.

```

%Calculamos el tiempo
for i=1:length(y)
    if y(2,i) <= 0.5*y0(2)
        fprintf('El tiempo para que se reduzca al 50 por ciento la
concentración de viriones es = %4.2f días', t(i));
        break
    end
end
end

```

El tiempo para que se reduzca al 50% la concentración de viriones es de 0.30 días

d) Interprete las gráficas obtenidas desde el punto de vista biológico.

Como se puede observar, el fármaco es efectivo, ya que realmente disminuye el número de células infectadas (gráfica de la izquierda) al disminuir su capacidad infecciosa. Esto se muestra también en las gráficas central y derecha, en las cuales se observa que el número de viriones tiende a 0 y el tiempo que tarda en anularse coincide con el tiempo en el cual se alcanza el máximo de partículas virales pero sin capacidad infecciosa en el organismo.

Práctica 7

**INTEGRACIÓN NUMÉRICA DE PROBLEMAS
DE VALOR INICIAL**

Apellidos, Nombre:

Apellidos, Nombre:

Grupo:

Ejercicio 1**Soluciones**

- a) Incluya la imagen resultante del perfil de concentraciones.
- b) ¿Cuál es la concentración final (último valor almacenado en el vector) en cada método? (Ayuda: los resultados quedan almacenados en el vector y)
- c) Ahora modifique el valor de h a 0.05. ¿Qué diferencias observa en la gráfica y en los valores de las concentraciones finales?
- d) Si el resultado exacto de la concentración final es 5.8121934843, ¿qué conclusiones puede extraer? ¿Cuál es el método que mejor aproxima la solución exacta en este caso?

Ejercicio 2**Soluciones**

- a) Incluya las tres subgráficas resultantes: C^* , V^* y V_N .
- b) Calcule el tiempo que se tarda en reducir a un 50% el número de viriones.
- c) Interprete las gráficas obtenidas desde un punto de vista biológico.

**Espacio para el
profesor**

Práctica 8

Ecuaciones en derivadas parciales (EDPs)

Aplicaciones biomédicas

- 🔥 Desintegración radiactiva
- 🔥 Curva disociación de la hemoglobina

Práctica 8

ECUACIONES EN DERIVADAS PARCIALES

Objetivos

El objetivo de esta práctica es entender y aplicar, de forma práctica, lo aprendido sobre resolución de ecuaciones en derivadas parciales (EDP) en las clases de teoría. Como en ellas, nos centraremos en EDPs lineales de tipo parabólico, si bien la técnica de diferencias finitas es también aplicable a ecuaciones de tipo hiperbólico y elíptico, e incluso a EDPs no lineales.

Actividades previas

Antes de la sesión **es necesario que:**

- Lea y comprenda la práctica. De esta manera aprovechará de manera más eficiente las horas de laboratorio.
- Realice los ejercicios señalados como **actividad previa**.

Nota previa

Las respuestas a las cuestiones planteadas, gráficos de soluciones, etc. deben rellenarse y entregarse en el cuaderno de entrega de soluciones disponible en el Moodle de la asignatura.

Ejercicio 1

Aplicación biomédica: Difusión simple de moléculas a través de una membrana plana

Planteamiento del problema

El mecanismo de difusión de moléculas a través de una membrana plana, como ocurre en la absorción de medicamentos a través de la piel, se puede modelar como

$$\frac{\partial C_A}{\partial t} = D_m \frac{\partial^2 C_A}{\partial x^2}, \quad 0 < x < L, t > 0,$$

donde C_A es la concentración del componente A en la membrana, D_m es el coeficiente de difusión del componente A en la membrana y x es la dirección de difusión. Se trata de un problema de valores iniciales y de contorno con una EDP parabólica unidimensional en el espacio. Las condiciones de contorno e iniciales a aplicar son las siguientes:

$$\begin{aligned} C_A(0, t) &= 1, t > 0, \\ C_A(L, t) &= 0, t > 0, \\ C_A(x, 0) &= 0, 0 \leq x \leq L. \end{aligned}$$

Desarrollo de la práctica

En la práctica, por simplicidad, utilizaremos C para denotar a la función incógnita (concentración) C_A .

- e) Codifique un m-file de función, de nombre GRxx_P8_E1.m, con las variables de salida x , t , C (en este orden) y con las siguientes variables de entrada:

- L = Distancia total de migración (cm). En nuestro caso, $L = 0.2$ cm.
- T = Máximo tiempo de integración (s). En nuestro caso, $T = 0.4$ s.
- N = Número de subintervalos de la malla espacial (variable x). En nuestro caso, $N = 100$.
- M = Número de subintervalos de la malla temporal (variable t). En nuestro caso, $M = 100$.
- D_m = Coeficiente de difusión (cm^2/s). En nuestro caso, $D_m = 10^{-4} \text{ cm}^2/\text{s}$.
- cond_inicial = Concentración del componente A en la membrana en el instante inicial.
- cc_izda = Concentración del componente A en la membrana en el contorno de la izquierda
- cc_dcha = Concentración del componente A en la membrana en el contorno de la derecha.

NOTA: En un caso general, las condiciones iniciales y de contorno se definirían mediante m-files de función separados (puesto que, en general, las condiciones iniciales pueden ser función de x , y las condiciones de contorno pueden ser funciones de t). En este ejercicio, sin embargo, como ambas condiciones son constantes las definiremos por simplicidad en el propio m-file del procedimiento de cálculo.

La secuencia de operaciones debe ser la siguiente:

- 1) Calcule las longitudes de paso espacial, h , y temporal, k , a partir de los valores de L , T , N y M .
- 2) Defina los vectores x y t , calculando los valores de sus componentes de acuerdo con los valores de h y k . En cuanto a la matriz C , para conseguir una mayor eficiencia computacional preasigne espacio de memoria usando para ello la función **zeros** (Nota: tenga en cuenta las dimensiones de la matriz).
- 3) Defina las constantes necesarias que aparezcan en el algoritmo del método explícito clásico. Recuerde que, para la ecuación de difusión no adimensionalizada que contiene la difusividad a , se define $r = ak/h^2$ (note que, en nuestro caso, D_m juega el papel de a).
- 4) Introduzca las condiciones iniciales en la matriz solución en el nivel temporal $t=0$.
- 5) Introduzca las condiciones de contorno de tipo Dirichlet en la matriz, tanto en el contorno izquierdo como en el derecho, pero con la precaución de preservar en la primera fila de la matriz las condiciones iniciales ya introducidas en el punto anterior.
- 6) Usando un doble bucle **for** (anidado), calcule los valores de C en todos los nodos internos de acuerdo con el algoritmo del método explícito clásico.

IMPORTANTE: Nótese que los índices de los vectores y matrices en Matlab empiezan en 1 (0 no está permitido). Nótese también que, en los métodos de diferencias finitas, C_{ij} denota $C(x_i, t_j)$, utilizándose habitualmente para x la coordenada horizontal y para t la vertical, es decir, i está asociado a columnas y j a filas. En Matlab, sin embargo, la notación está invertida respecto a la anterior, pues con $C(i,j)$ nos referimos al elemento de la matriz C en la fila i y la columna j . En consecuencia, los valores de C_{ij} calculados deben almacenarse en las posiciones $C(j,i)$.

- 7) Dibuje la solución numérica de 5 perfiles de concentración normalizados para así observar cómo se produce la difusión. Para ello, copie el siguiente fragmento de código:

```
s=5;
new_C(1,:) = C(1,:)/cc_izda;
cuenta = fix((length(t)-1)/s);
for p=1:s
    new_C(p+1,:) = C(1+p*cuenta,:)/cc_izda;
end
clf; figure(1); plot(x,new_C);
```

Añada los nombres de los ejes y un título para la figura, y limite, mediante un mandato **axis**, los valores de x representados a valores entre 0 y 0.05 (los del eje y puede fijarlos entre 0 y 1).

Desde la ventana de mandatos, llame a ejecución a la función, con los valores de las variables de entrada especificados más arriba. Para que los vectores x y t y la matriz C queden almacenados en memoria, preceda la llamada a la función por “[x,t,C]=”.

- f) ¿Qué valor de r se ha obtenido en el caso anterior? Recuerde que $r=ak/h^2$
- g) Modifique ahora los valores de L y T por los siguientes: $L=0.1\text{cm}$, $T=0.8\text{s}$, manteniendo el resto de parámetros, pero excluya la limitación de los valores de x introducida en el apartado (a) mediante el mandato **axis**. ¿Qué ocurre ahora? ¿Cuál es el motivo?

Entrega de resultados

Una vez terminado el ejercicio, cree una carpeta comprimida de nombre GRxx_P8.zip, en la que debe incluir el m-file y el documento Word de respuestas debidamente cumplimentado (incluyendo las gráficas obtenidas), y suba este fichero al Moodle de la asignatura.

Práctica 8

ECUACIONES EN DERIVADAS PARCIALES

Objetivo de la práctica

El objetivo de esta práctica es entender y aplicar, de forma práctica, lo aprendido sobre resolución de ecuaciones en derivadas parciales (EDP) en las clases de teoría. Como en ellas, nos centraremos en EDPs lineales de tipo parabólico, si bien la técnica de diferencias finitas es también aplicable a ecuaciones de tipo hiperbólico y elíptico, e incluso a EDPs no lineales.

Ejercicio 1

Soluciones

a) Codifique un m-file de función, de nombre GRxx_P8_E1.m.

```
function [x,t,C] = GRxx_P8_E1(L,T,N,M,Dm,cond_inicial,cc_izda,cc_dcha)

% L = Distancia total de migración(cm)
% T = Máximo tiempo de integración(s)
% N = Número de subintervalos de malla espacial (variable x)
% M = Número de subintervalos de malla temporal (variable t)
% Dm = Coeficiente de difusión (cm^2/s)
% cond_inicial = Concentración del componente A en la membrana
% en el instante inicial
% cc_izda = Concentración del componente A en la membrana en
% el contorno de la izquierda
% cc_dcha = Concentración del componente A en la membrana en
% el contorno de la derecha

% Calculamos las longitudes de paso espacial y temporal
% y definimos los vectores x y t
h = L/N;
x = ([0:N]*h); % o bien x=0:h:L

k = T/M;
t = ([0:M]*k); ; % o bien t=0:k:T

% Calculamos las constantes
r = Dm*(k/h^2);

% Creamos la malla
C = zeros(M+1,N+1);

% Introducimos las condiciones iniciales y de contorno en la
% malla
C(2:M+1,1) = cc_izda;
C(2:M+1,N+1) = cc_dcha;
C(1,:)= cond_inicial;

% Lo que sigue sería introduciendo primero las condiciones de % contorno para
% todo t y luego las iniciales sólo en los nodos % internos (la gráfica resulta
% más "fea")
% C(1:M+1,1) = cc_izda;
% C(1:M+1,N+1) = cc_dcha;
% C(1,2:N)= cond_inicial;

% Resolvemos
% Iteramos en t (bucle externo) y en x (bucle interno)
for j = 1:M
    for i=2:N
```

```

        C(j+1,i) = r*C(j,i+1) + (1-2*r)*C(j,i) + r*C(j,i-1);
    end
end

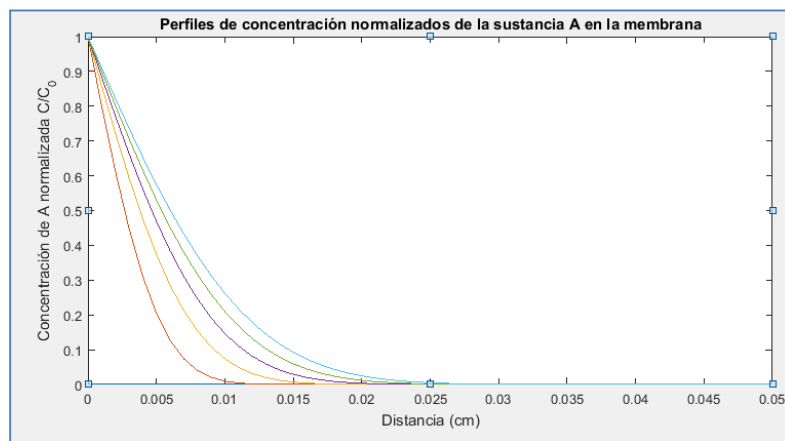
% Dibujamos la solución numérica
s=5;
new_C(1,:) = C(1,+)/cc_izda;
cuenta = fix((length(t)-1)/s);
for p=1:s
    new_C(p+1,:) = C(1+p*cuenta,+)/cc_izda;
end
clf; figure(1); plot(x,new_C);
axis([0 0.05 0 1])
xlabel('Distancia (cm)')
ylabel('Concentración de A normalizada C/C_0')
title('Perfiles de concentración normalizados de la sustancia A en la membrana')

```

Llamando la función a ejecución con el siguiente mandato

`[x,t,C]=GRxx_P8_E1(0.2,0.4,100,100,1e-4,0,1,0)`

se obtiene la siguiente figura:

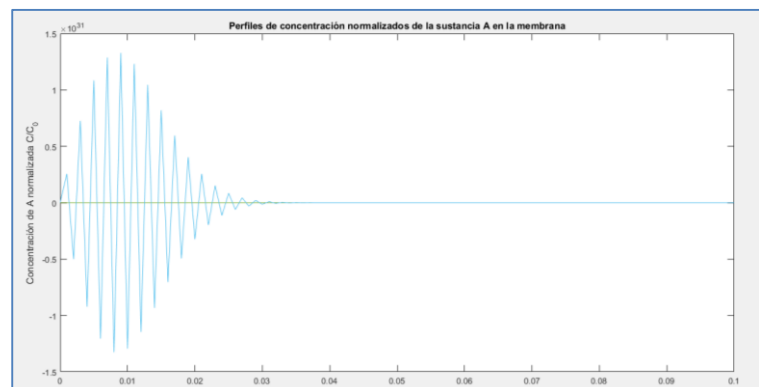


b) ¿Qué valor de r se ha obtenido en el caso anterior?

El valor de r viene dado por $r = ak/h^2$. Sabiendo que en este caso a es Dm , $k = T/M$ y $h = L/N$, sustituyendo los valores dados obtenemos $r=0.1$.

c) Modifique ahora los valores de L y T por los siguientes: $L=0.1\text{cm}$, $T=0.8\text{s}$, manteniendo el resto de parámetros, pero excluya la limitación de los valores de x introducida en el apartado (a) mediante el mandato **axis**. ¿Qué ocurre ahora? ¿Cuál es el motivo?

El valor obtenido ahora para r es 0.8, de modo que no se cumple la condición de estabilidad $r \leq 0.5$ y la solución es inestable, comenzando a oscilar como se muestra en la siguiente figura:



Práctica 8

ECUACIONES EN DERIVADAS PARCIALES

Apellidos, Nombre:

Apellidos, Nombre:

Grupo:

Ejercicio 1**Soluciones**

- a) Incluya la gráfica resultante de la ejecución de la función GRxx_P8_E1.m para los valores de las variables de entrada dados.
- b) ¿Qué valor de r se ha obtenido en el caso anterior?
- c) Modifique ahora los valores de L y T por los siguientes: $L=0.1\text{cm}$, $T=0.8\text{s}$, manteniendo el resto de parámetros, pero excluya la limitación de los valores de x introducida en el apartado (a) mediante el mandato **axis**. Incluya la gráfica resultante en este caso. ¿Qué ocurre ahora? ¿Cuál es el motivo?

**Espacio para el
profesor**

Bibliografía

- S. M. Dunn, A. Constantinides, P. V. Moghe. Numerical methods in Biomedical Engineering. *Elsevier Academic Press*, 2006
- C. Cobelli, E. Carson. Introduction to Modelling in Physiology and Medicine. *Elsevier Academic Press*, 2008
- M. R. King, N. A. Mody. Numerical and Statistical Methods for Bioengineering. Applications in MATLAB. *Cambridge University Press*.
- A. T. Layton, G. Wei. Interface methods for biological and biomedical problems. *International Journal for Numerical Methods in Biomedical Engineering*, pp. 289-290. 2012
- J. C. Maldonado Beltrán, Ronald Gerardo García Gómez, Lola Xiomara Bautista Roza. Matlab application for the estimation of Blood Pressure Variability. *Gerenc. Tecnol. Inform*, Vol. 11, N° 30, pp. 57-65 2012.
- D. L. Nelson. Lehninger: Principios de Bioquímica. 5ª ed. *Omega*, 2007
- J.E. Hall , Guyton y Hall. Tratado De Fisiología Médica. *S.A. Elsevier España*, 2011
- G. W. Recktenwald. Numerical Methods with Matlab: Implementations and Applications. *Prentice-Hall*, 2003
- J. H. Mathews, K. D. Fink. Numerical Methods Using Matlab. *Prentice Hall*. 1999.
- A. Kharab, R. B. Guenther. An Introduction to Numerical Methods: A MATLAB Approach, Third Edition. *Chapman and Hall/CRC*. 2011
- Faires, J. Douglas. Métodos numéricos. 3ª ed. *Thomson-Paraninfo*, 2004
- J. D. Lambert, Numerical methods for ordinary differential systems: the initial value problem. *John Wiley & Sons*, 1991
- Quarteroni, Alfio. Scientific computing with MATLAB and Octave. *Springer*, 2010
- Oomens, Cees. Biomechanics: concepts and computation. *Cambridge University Press*, 2009
- Burden, Richard L. Análisis numérico. 7ª ed. *International Thomson*, 2002.
- Chemical Engineers in the Biomedical Field: from discoveries to solutions. *American Institute of Chemical Engineers and Chemical Heritage Foundation*. 2009

